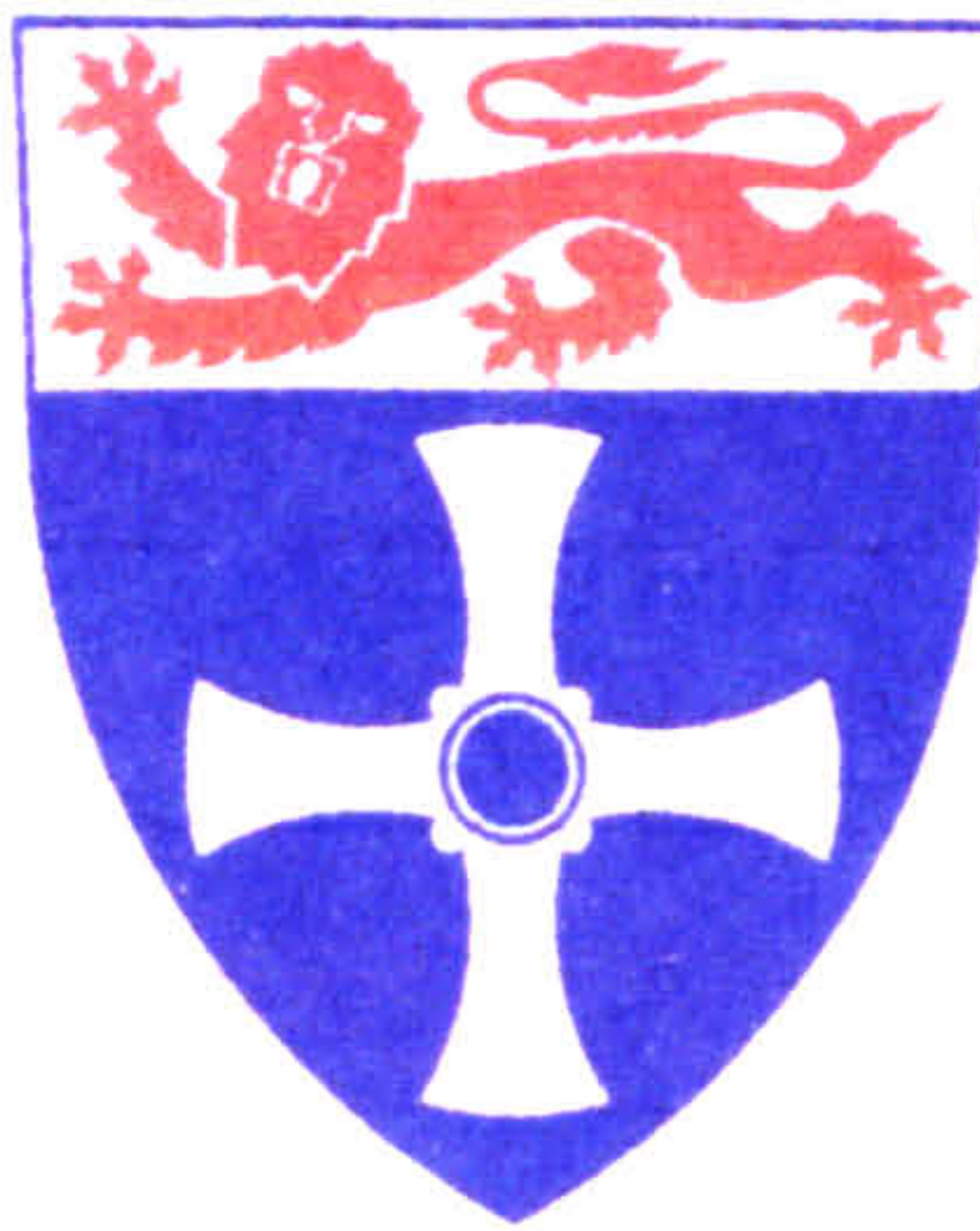


Methodologies for Distributed and Higher Dimensional Geographic Information

UNIVERSITY OF
NEWCASTLE UPON TYNE



Scott Parsley BSc. (Hon)

**Thesis Submitted for the Degree of Doctor
of Philosophy**

**Department of Geomatics
University of Newcastle upon Tyne**

June 2001

NEWCASTLE UNIVERSITY LIBRARY

200 25041 2

Thesis L6929

“Taking mapping to infinity and beyond”

Scott Parsley (12 January 2001)

“Ours is not to reason why, ours is but to do and die”

Alfred Lord Tennyson

Abstract

In today's digital era, cartography has changed its role, from that of a pure visual model of the Earth's surface, to an interface to other spatial and aspatial information. Along with this, representation and manipulation of graphical information in three-dimensional space is required for many applications. Problems and difficulties must be overcome in order to facilitate the move to three-dimensional models, multimedia, and distributed data. Can accurate measurements, at sufficient resolution, and using affordable resources be obtained? Will application software usefully process, in all aspects, models of the real world, sounds, and videos? Combined with this, the workplace is becoming distributed, requiring applications and data that can be used across the globe as easily as in the office.

A distributed, three-dimensional, GIS is required with all the procedural and recording functionality of current two-dimensional systems. Such a GIS would maintain a model, typically comprised of solids of individual buildings, roads, utilities etc. with both external and internal detail, represented on a suitable digital terrain model. This research examines virtual reality software as part of an answer. Alternatively, can technologies such as HTML, VRML, and scripting, along with object-orientation and open systems, allow for the display and interrogation of networked data sets?

The particular application of this technology, considered during this research, is the need for accurate reconstruction of historical urban monuments. The construction, manipulation, and exploration of these models is often referred to as virtual heritage. This research constructs an innovative and resource effective methodology, the Phoenix algorithm, which requires only a single image for creating three-dimensional models of buildings at large scale. The development of this algorithm is discussed and the results obtained from it are compared with those obtained using traditional three-dimensional capture techniques. Furthermore, possible solutions to the earlier questions are given and discussed.

Table of Contents

ABSTRACTI

TABLE OF CONTENTS..... II

LIST OF FIGURES..... V

LIST OF TABLES.....VII

LIST OF ACRONYMS AND ABBREVIATIONS..... VIII

ACKNOWLEDGEMENTS.....XI

1. INTRODUCTION 1

1.1 AIMS AND OBJECTIVES 2

1.2 BACKGROUND OF RESEARCH..... 2

1.3 OUTLINE OF THESIS 3

2. EVOLUTION OF SPATIAL INFORMATION 8

2.1 GEOGRAPHIC INFORMATION SYSTEMS 9

2.2 GEOGRAPHIC INFORMATION 14

2.3 MULTIMEDIA AND TEMPORAL DATA..... 17

2.4 THREE DIMENSIONALITY 21

2.5 VIRTUAL REALITY 25

2.6 THE INTERNET AND THE WORLD WIDE WEB..... 33

2.7 SUMMARY 35

3. APPLYING WEB TECHNOLOGY 37

3.1 HYPERTEXT MARKUP LANGUAGE..... 39

3.2 DYNAMIC HTML 42

3.2.1 *Style sheets* 43

3.2.2 *Document Object Model* 45

3.2.3 *Scripting*..... 46

3.3 DISPLAYING GRAPHICS IN HTML FILES 47

3.4 VIRTUAL REALITY MODELLING LANGUAGE..... 50

3.5 FUTURE DIRECTIONS 54

3.5.1 *Element Definition*..... 58

3.5.2 *Styles*..... 59

3.5.3 *Linking*..... 60

3.6 VECTOR MARK-UP LANGUAGES..... 62

3.7 SUMMARY 64

4. COMPONENT SYSTEMS..... 66

4.1 NETWORKING TECHNOLOGY..... 67

4.2 OBJECT-ORIENTATION AND COMPONENTS 72

4.3 SPATIAL OPERATOR STANDARDS 74

4.4 APPLICATION OBJECT STANDARDS 76

4.4.1 *The Object Management Group*..... 79

4.4.2 *Object Data Management Group*..... 81

4.4.3 *The Open GIS Consortium*..... 83

4.4.4 *Microsoft*..... 85

4.5 QUERYING DATA..... 89

4.6 SUMMARY 92

5. DATA STRUCTURE 94

5.1	TWO-DIMENSIONAL DATA STRUCTURES.....	95
5.1.1	Two-dimensional Raster formats.....	96
5.1.2	Example Binary Vector Format.....	97
5.1.3	Fractals.....	100
5.2	THREE-DIMENSIONAL DATA STRUCTURES.....	102
5.2.1	Vector Structures.....	103
5.2.2	Raster Structures.....	107
5.2.3	Three-dimensional Primitives.....	108
5.2.4	Stereograms.....	109
5.2.5	QuickTime VR.....	114
5.3	FOUR-DIMENSIONAL DATA STRUCTURES.....	115
5.4	OBJECT-ORIENTED AND HIERARCHICAL STRUCTURES.....	118
5.5	GENERALISATION AND LEVELS OF DETAIL.....	120
5.6	APPLICATION OF A VR DATA STRUCTURE TO GIS DATA.....	124
5.7	SUMMARY.....	126
6.	THREE-DIMENSIONAL DATA CAPTURE.....	128
6.1	EXTENSION OF EXISTING DATA.....	130
6.2	THEODOLITE SURVEY.....	133
6.3	LASER RANGING SYSTEMS.....	135
6.4	SATELLITE POSITIONING.....	138
6.5	STEREOSCOPIC METHODS.....	139
6.6	LIGHT METHODS.....	144
6.7	MULTIPLE OBLIQUE IMAGERY.....	147
6.7.1	Shape from video imagery.....	149
6.8	SUMMARY.....	149
7.	SINGLE IMAGE METHODS.....	151
7.1	PERSPECTIVE GEOMETRY.....	151
7.2	MATHEMATICS OF PROJECTION.....	154
7.3	VANISHING POINTS.....	156
7.4	GRAPHICAL TECHNIQUES.....	158
7.5	AUTOMATED METHODS.....	163
7.5.1	Labelled line drawings.....	164
7.5.2	Gaussian Mapping.....	165
7.6	SUMMARY.....	166
8.	PHOENIX ALGORITHMS.....	168
8.1	BACKGROUND.....	168
8.1.1	Kodak Dimension, PhotoModeler and Canoma.....	169
8.1.2	Mathematical notation.....	170
8.2	PRINCIPAL POINT.....	171
8.3	PHOENIX V1.....	173
8.4	PHOENIX V2.....	178
8.5	SUMMARY.....	186
9.	TESTING OF PHOENIX.....	187
9.1	TEST METHODS.....	187
9.1.1	Theodolite angle intersection.....	189
9.1.2	Metric photogrammetry.....	190
9.1.3	PhotoModeler.....	190
9.1.4	Phoenix approach.....	191
9.2	TEST ONE RESULTS.....	192
9.2.1	Location of vanishing points.....	195
9.3	POLICE HOUSE.....	197
9.3.1	Results of second test.....	200
9.4	DISCUSSION OF ERRORS IN PHOENIX APPROACH.....	202
9.5	HISTORIC MODEL.....	203
9.6	SUMMARY.....	206

10. CONCLUSIONS..... 207

10.1 DISPLAY OF GEOGRAPHIC INFORMATION..... 208

10.2 THE INTERNET 211

10.3 COMPONENT TECHNOLOGY 213

10.4 MOVING FROM TWO TO THREE DIMENSIONS..... 216

10.5 STRUCTURING DATA FOR THE THIRD DIMENSION..... 219

10.6 GATHERING INFORMATION FROM SINGLE IMAGES 220

10.7 FUTURE WORK 223

REFERENCES 225

TABLE OF APPENDICES 231

APPENDIX 1: SURVEY DATA..... 232

List of Figures

FIGURE 2.1: MOORE'S LAW.....	9
FIGURE 2.2: THE COMPONENTS OF VISUALISATION.....	11
FIGURE 2.3: TAYLOR'S COMPONENTS OF VISUALISATION ADAPTED TO CURRENT SITUATION FOR GIS	13
FIGURE 2.4: IMAGES FOUND WHEN SEARCHING FOR +PHOENIX +BIRD.....	19
FIGURE 2.5: EXAMPLE OF VISUALLY SIMILAR IMAGES	20
FIGURE 2.6: THE EVOLUTION OF THE MAPPING OF LAND SURFACE FORM.....	22
FIGURE 2.7: CLASSIFICATION OF TECHNIQUES FOR VISUALISING TERRAIN.....	23
FIGURE 2.8: VIRTUAL WORLDS CREATED FROM ACCURATE MAPPING DATA.....	24
FIGURE 2.9: THE CYBERSPHERE - A FULLY IMMERSIVE SPHERICAL PROJECTION SYSTEM.....	28
FIGURE 2.10: GIS PROJECTS USING VIRTUAL REALITY	32
FIGURE 3.1: SCREENSHOT OF VIRTUAL HISTORIC NEWCASTLE PROTOTYPE	38
FIGURE 3.2: BASIC STRUCTURE OF AN HTML FILE.....	40
FIGURE 3.3: EXAMPLE OF A CASCADING STYLE SHEET.....	43
FIGURE 3.4: RESULTING OUTPUT OF A CASCADING STYLE SHEET	44
FIGURE 3.5: THE INTERNET EXPLORER 4.0 DOCUMENT OBJECT MODEL.....	45
FIGURE 3.6: EXAMPLE OF TAG FORMAT.....	48
FIGURE 3.7: EXAMPLE CLIENT-SIDE IMAGE MAP.....	49
FIGURE 3.8: EXAMPLE HTML AND RELATED DISPLAY	56
FIGURE 3.9: EXAMPLE OF A XML FILE.....	57
FIGURE 3.10: PART OF THE DTD FOR THE EARLIER XML DOCUMENT.....	58
FIGURE 3.11: PART OF THE XSL FOR THE ABOVE XML DOCUMENT.....	59
FIGURE 3.12: EXAMPLE OF APPLIED XSL.....	60
FIGURE 3.13: EXAMPLE VML OF A STAR SHAPE.....	63
FIGURE 4.1: NSFNET TRAFFIC	69
FIGURE 4.2: THE MIGRATION FROM TRADITIONAL GIS	76
FIGURE 4.3: OBJECT MODEL ARCHITECTURE	78
FIGURE 4.4: THE UNIVERSAL DATA ACCESS ARCHITECTURE.....	87
FIGURE 5.1: THE ENHANCEMENT OF A SHAPE USING FRACTALS.....	101
FIGURE 5.2: DATA STRUCTURES FOR DTM DEFINITION.....	104
FIGURE 5.3: A SINGLE IMAGE RANDOM TEXT STEREOGRAM.....	110
FIGURE 5.4: PRINCIPLE OF THREE-DIMENSIONAL DATA STORAGE IN SIRTS	111
FIGURE 5.5: A SINGLE IMAGE RANDOM DOT STEREOGRAM	112
FIGURE 5.6: IMAGE USED TO GENERATE PREVIOUS SIRDS.....	113
FIGURE 5.7: QUICKTIME VR MODEL OF WESTMINSTER ABBEY.....	114
FIGURE 5.8: A STEREOPAIR OF A FOUR-DIMENSIONAL HYPERCUBE	117
FIGURE 5.9: EXAMPLE OF LEVELS OF DETAIL OF AN OBJECT.....	121
FIGURE 5.10: VRML MODEL OF A SHIP BEING GENERALISED BY LODSTAR	122
FIGURE 6.1: USAGE OF AN OBJECT'S SKELETON TO CREATE A ROOFLINE	132
FIGURE 6.2: USE OF EXTENDING EXISTING DATA TO PRODUCE A 3D TOWNSCAPE.	133
FIGURE 6.3: THE MODELMAKER SYSTEM.....	136
FIGURE 6.4: LIDAR DATA FOR NEWCASTLE UPON TYNE	137
FIGURE 6.5: IMAGERY OF SAN FRANCISCO, CALIFORNIA FROM THE IKONOS-1 SATELLITE.....	142
FIGURE 6.6: A MOIRÉ PROJECTION SYSTEM	145
FIGURE 6.7: MOIRÉ FRINGE PATTERNS	145
FIGURE 6.8: MODEL CREATION BY THE INSPECK-3D SYSTEM.....	146
FIGURE 7.1: 'ASCENDING AND DESCENDING'.....	152
FIGURE 7.2: HAYWARD'S 'UNDECIDABLE MONUMENT'	153
FIGURE 7.3: PARALLEL PROJECTIVE TRANSFORM	155
FIGURE 7.4: GEOMETRY OF THE EYE (A), THE CAMERA (B) AND PERSPECTIVE (C).....	155
FIGURE 7.5: LOCATION OF VANISHING POINTS OF AN OBJECT.....	157
FIGURE 7.6: THREE POINT PERSPECTIVE GEOMETRY	160
FIGURE 7.7: GENERATION OF PLAN VIEW	162
FIGURE 7.8: GENERATION OF ELEVATION.....	163
FIGURE 7.9: AN EXAMPLE OF A LABELLED LINE DRAWING	165
FIGURE 8.1: GRAPHICAL CREATION OF THE PRINCIPAL POINT, P FOR A CROPPED IMAGE.....	172

FIGURE 8.2: USER INPUT FOR PHOENIX V1	174
FIGURE 8.3: GEOMETRY FOR PHOENIX V1	175
FIGURE 8.4: PERSPECTIVE TRIANGLE.....	179
FIGURE 8.5: VIEW OF CO-ORDINATE SYSTEMS IN ELEVATION (A) AND PLAN (B).....	183
FIGURE 9.1: MODERN DAY TEST BUILDING.....	188
FIGURE 9.2: COMPARISONS OF MODELS FROM TEST ONE.....	193
FIGURE 9.3: RANGE OF MODELS PRODUCED FROM DIFFERING VANISHING POINTS.....	196
FIGURE 9.4: LOCATION OF POLICE HOUSE SURVEYED	198
FIGURE 9.5: SINGLE IMAGE USED FOR MODEL OF POLICE HOUSE	199
FIGURE 9.6: COMPARISONS OF PHOENIX AND SURVEYED MODELS	200
FIGURE 9.7: HISTORIC IMAGE PROCESSED BY PHOENIX.....	204
FIGURE 9.8: VR MODEL CREATED FROM HISTORIC IMAGE.....	205

List of Tables

TABLE 4.1: ROUTE TO WWW.ESRI.COM FROM BOWDITCH.NCL.AC.UK.....	71
TABLE 4.2: DATA STRUCTURES	75
TABLE 5.1: SPF DATA FORMAT.....	99
TABLE 5.2: THE RESULTS FOR A MODEL GENERALISED BY LODESTAR.....	122
TABLE 6.1: COMPARISON OF ERRORS FOR STAND-ALONE AND DIFFERENTIAL GPS.....	138
TABLE 6.2: SUMMARY OF EO DATA CHARACTERISTICS.....	143
TABLE 9.1: SUMMARY OF DIFFERENCES OF MODELS WITH BASE MODEL.....	194
TABLE 9.2: RESULTS OF STATISTICAL TESTS ON MODELS FROM TEST TWO.....	201

List of Acronyms and Abbreviations

2D TWO-DIMENSIONAL
3D THREE-DIMENSIONAL

A

ADO ACTIVE X DATA OBJECTS
ADT ABSTRACT DATA TYPE
AGOCG ADVISORY GROUP ON COMPUTER GRAPHICS
AI ARTIFICIAL INTELLIGENCE
ANSI AMERICAN NATIONAL STANDARDS INSTITUTE
API APPLICATION PROGRAMMING INTERFACE
ARPA ADVANCED RESEARCH PROJECTS AGENCY

C

CAD COMPUTER AIDED DESIGN
CAE COMPUTER AIDED ENGINEERING
CAM COMPUTER AIDED MANUFACTURING
CEN CONSEIL EUROPÉEN POUR LA NORMALISATION
CGIS CANADA GEOGRAPHIC INFORMATION SYSTEM
COM COMPONENT OBJECT MODEL
CORBA COMMON OBJECT REQUEST BROKER ARCHITECTURE
CSS CASCADING STYLE SHEETS

D

DAO DIRECT ACCESS OBJECTS
DB DATABASE
DBMS DATABASE MANAGEMENT SYSTEM
DDE DYNAMIC DATA EXCHANGE
DEM DIGITAL ELEVATION MODEL
DHTML DYNAMIC HYPERTEXT MARKUP LANGUAGE
DIGEST DIGITAL GEOGRAPHIC STANDARD
DLL DYNAMIC LINK LIBRARY
DNA DISTRIBUTED INTERNET APPLICATIONS
DOM DOCUMENT OBJECT MODEL
DPI DOTS PER INCH
DTD DOCUMENT TYPE DECLARATION
DTM DIGITAL TERRAIN MODEL
DXF DIGITAL EXCHANGE FORMAT
DVD DIGITAL VERSATILE DISK

E

EDM ELECTRONIC DISTANCE MEASUREMENT
EO EARTH OBSERVATION
ESRI ENVIRONMENTAL RESEARCH SYSTEMS INSTITUTE

F

FTP FILE TRANSFER PROTOCOL

G

GB GIGABYTE
GIF GRAPHICS INTERCHANGE FORMAT
GIS GEOGRAPHICAL INFORMATION SYSTEM
GPS GLOBAL POSITIONING SYSTEM

H

HITL	HUMAN INTERFACE TECHNOLOGY LABORATORY
HTML	HYPERTEXT MARKUP LANGUAGE

I

ICS	IMAGE CO-ORDINATE SYSTEM
IHO	INTERNATIONAL HYDROGRAPHIC ORGANISATION
IIS	INTERNET INFORMATION SERVER
ISO	INTERNATIONAL STANDARDS ORGANISATION

J

JANET	JOINT ACADEMIC NETWORK
JARIC	JOINT AIR RECONNAISSANCE INTELLIGENCE CORPS
JPEG	JOINT PHOTOGRAPHIC EXPERTS GROUP

L

LIDAR	LASER INDUCED DIRECTION AND RANGE SYSTEM
LOD	LEVEL OF DETAIL
LZW	LEMPERL-ZIV-WELCH

M

MB	MEGABYTE
MHz	MEGAHERTZ
MS	MICROSOFT

N

NSF	NATIONAL SCIENCE FOUNDATION
NTF	NATIONAL TRANSFER FORMAT

O

OCS	OBJECT CO-ORDINATE SYSTEM
ODBC	OPEN DATABASE CONNECTIVITY
ODBMS	OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEM
ODL	OBJECT DEFINITION LANGUAGE
ODMG	OBJECT DATA MANAGEMENT GROUP
OGC	OPEN GIS CONSORTIUM
OGIS	OPEN GEODATA INTEROPERABILITY SPECIFICATION
OGM	OGIS GEODATA MODEL
OLE	OBJECT LINKING AND EMBEDDING
OMA	OBJECT MANAGEMENT ARCHITECTURE
OMG	OBJECT MANAGEMENT GROUP
ORM	OGIS REFERENCE MODEL
OQL	OBJECT QUERY LANGUAGE
ORB	OBJECT REQUEST BROKER
OS	ORDNANCE SURVEY

P

PC	PERSONAL COMPUTER
----	-------------------

Q

QTM	QUATERNARY TRIANGULAR MESH
-----	----------------------------

R

RDO	REMOTE DATA OBJECTS
RDS	REMOTE DATA SERVICE
ROM	READ ONLY MEMORY

S

SAIF	SPATIAL ARCHIVE INTERCHANGE FORMAT
SDTS	SPATIAL DATA TRANSFER STANDARD
SGML	STANDARD GENERALISED MARKUP LANGUAGE
SIRDS	SINGLE IMAGE RANDOM DOT STEREOGRAM
SIRTS	SINGLE IMAGE RANDOM TEXT STEREOGRAM
SQL	STRUCTURED QUERY LANGUAGE
SQL3-MM	STRUCTURED QUERY LANGUAGE 3 - MULTIMEDIA
SVG	SCALABLE VECTOR GRAPHICS

T

TCP/IP	TRANSMISSION CONTROL PROTOCOL / INTERNET PROTOCOL
TELNET	TERMINAL EMULATION
TIFF	TAGGED IMAGE FILE FORMAT

V

VHN	VIRTUAL HISTORIC NEWCASTLE
VHR	VERY HIGH RESOLUTION
VML	VECTOR MARKUP LANGUAGE
VR	VIRTUAL REALITY
VRGIS	VIRTUAL REALITY GEOGRAPHIC INFORMATION SYSTEM
VRML	VIRTUAL REALITY MODELLING LANGUAGE

W

W3C	WORLD WIDE WEB CONSORTIUM
WEB	WORLD WIDE WEB
WWW	WORLD WIDE WEB
WYSIWYG	WHAT YOU SEE IS WHAT YOU GET

X

XML	EXTENSIBLE MARKUP LANGUAGE
XSL	EXTENSIBLE STYLE LANGUAGE

Acknowledgements

In all honesty, far too many people should really be included that I am giving up trying to remember them all. Also there is the factor that if I did then this would probably become the longest Section of this thesis. So instead, here are the main ones.

First is Kate Willars, who has been many things to me during this PhD. Most importantly though she has convinced me to stay with it on the many times I have nearly given up.

Then most definitely there has to be George Taylor for his words of wisdom as a supervisor, prods in the right direction when I looked lost and bullying me into writing up.

Geoff Twentyman and the team at JARIC (Joint Air Reconnaissance Intelligence Corps) aided most of the work on single images. With special thanks to Cy Kelly and the Canadian Department of National Defence for giving me permission to use their manuals on graphical techniques for making measurements and models from single images.

Thanks also go to Guy Pearson and Ann Bottril for their work done in testing of the Phoenix algorithm during the summer of 1998, and to Dave Heslop supplied the photographs and some of the ideas towards the creation of Virtual Historic Newcastle a prototype of which formed some of this PhD.

My parents of course supported me as best they could considering they haven't seen much of me in the four years it took to produce this.

A big thanks to all the friends who have supplied nights out climbing, swimming, at the cinema, and of course at the pub, when it came time to escape from work for the sanity of my mind. Of course, also a big cheer for those who spared time for proof reading all this for me.

Finally, I have to thank Marks and Spencers for opening a massive food hall opposite the University. The food there has cheered me up on many a day when depressed with the workload.

1. Introduction

This research investigates the development of both current and new technologies to enhance geographic information and the applications that use it. Primarily concerned with the handling of large-scale geographic data, the research discusses approaches to converting and creating data sets accessible to the growing community of geographic information users. These aspects are discussed in relation to their application in creating a three-dimensional landscape of historic Newcastle upon Tyne that is accessible across the Internet, the Virtual Historic Newcastle (VHN) project. Of particular interest to the VHN project, the research also contributes a new method for collecting three-dimensional information from single oblique images, described as the Phoenix approach.

The graphical representation, and interactive manipulation of, accurate three-dimensional models of reality is rapidly becoming a necessity for many applications, especially those where the public is the final customer. Along with this, collaborative working methods and increased global co-operation has led to an increase in the quantities of distributed information. Networking technologies, particularly the Internet, have opened new directions for computing, allowing information to be shared across the world as easily as across the office. What is required from geographic information science is the technology to use these new data sets with all the procedural and analytical functionality of current systems. With the cost of computer hardware decreasing, yet power increasing, spatial information is evolving to include more data sources and types. These new technologies will allow digital spatial information to become three-dimensional, temporal, multi-scale, and contain hierarchical structure.

1.1 Aims and Objectives

The aims and objectives of the research are to

- Review current methods and technologies for displaying geographic data on the Internet.
- Discuss the structure of three-dimensional data for use in virtual reality and Internet applications.
- Examine the requirements and techniques for creating data sets that contain three and higher dimensions.
- Create a new methodology to extract three-dimensional information from single, oblique, images.

1.2 Background of Research

The background to the research, as stated above, is a requirement for a virtual reality (VR) townscape of Newcastle upon Tyne, the VHN project. The townscape is required to aid archaeologists in discovering more about Newcastle, as well as allowing locals and tourists to experience Newcastle as it was at key stages throughout its development. The VHN project is referred to throughout the chapters to show how the technology discussed is related to the aims and objectives of the thesis. The project has two main requirements, firstly the ability to access and interrogate the data across the Internet to allow collaborative working. Secondly, the ability to create three-dimensional models of objects and scenes that no longer exist. The subsequent implementation of the models in virtual reality software allows researchers to experiment with the world in ways not possible in real life.

1.3 Outline of Thesis

Chapter 2 examines the background and development of spatial data and applications, from their two-dimensional roots, to three-dimensional information that is distributed across a network, as required by the VHN project. Discussing topics including; multimedia, VR, and the World Wide Web (referred to hereafter as just the 'web'), it forms a brief introduction to these technologies that are expanded upon in later chapters. The chapter itself suggests that to move to three-dimensional and distributed information requires a change in attitude towards the applications, the information, and the users. To highlight this, the chapter gives an overview of how these technologies fit together to enable the VHN project, and gives brief descriptions of example applications that are already encompassing three-dimensional and distributed technologies.

The concepts of the Internet and the web have recently been at the forefront in the development of computing, with web addresses now appearing everywhere, for example on advertisements and after television programmes. Chapter 3 describes some of the applications of current and developing technologies behind the web in a geographic framework. Outlining technologies for the display of textual and graphical spatial data, the chapter documents the data formats enabling the technology. The chapter demonstrates using the Internet technology available during the research to implement aspects of the user interface to the VHN project. It then progresses to discuss the possibilities for moving away from the present two-dimensional web, by illustrating candidates allowing intelligent three-dimensional graphics and VR. Throughout, Chapter 3 illustrates how these technologies currently allow the distribution of geographical information, as well as how they need to develop further.

Chapter 4, examines the area of distributed computing in more detail and builds on the introduction in Chapter 2, to give a snapshot of current methods and technologies for component technology. By integrating the web interface discussed in Chapter 3 with underlying Internet and component technology, the VHN project has a means to distribute the data, applications, and the interface from the source to the user. The overview examines the networking technology used to transfer the data as well as the developing standards for naming objects and distributed software. Further examples of commercial software applications that have begun to implement these ideas in the field of spatial information are given.

Current approaches to the structure of data are discussed in Chapter 5. These include vector and raster methods, higher dimensional data (i.e. four-dimensional), and a new binary vector format developed as part of this research. This chapter details the pros and cons of current file formats, concepts for geographic information discussed in the applicability to the VHN project, and suggests and introduces new ideas for these data structures. These ideas include the two-dimensional encoding of three-dimensional images and the use of a simple raster based pseudo-VR. Through the construction of landscape and feature information for the VHN project, the differences between data formats for three-dimensional computer aided design (CAD) and VR are also highlighted, with a brief discussion on the problems of converting between the two.

Chapter 5 also examines how the concept of object-orientation and the related ideas of inheritance and encapsulation can be applied to spatial information in addition to software. The chapter then highlights how this approach can be used to benefit the structure and storage of large geographic data sets, including the application of a hierarchical data structure. This idea is then taken further for the VHN project to

include the addition of generalisation and levels of detail in the three-dimensional model used.

The discussion in Chapter 2 suggests that the development of three-dimensional geographical information systems (GIS) applications requires a change in the attitude towards the capture of spatial data. Chapter 6 discusses the necessity for this change in approach, examining techniques for the capture of both landscape and land-based features. Building on current techniques, including theodolite intersection and photogrammetry, the chapter expands to cover a sample of the new technologies on the market, and in research. This includes areas such as new high-resolution satellite imagery, laser induced direction and range system (LIDAR), and the use of video imagery to create large-scale three-dimensional data. The applicability of these techniques to the construction of the VHN project is discussed.

The final part of the research, Chapters 7, 8, and 9 deals with a specific problem of data capture raised from the VHN project. Due to the historic nature of the model, only single images exist for many of the buildings and features to be included. Therefore, a technique was required to capture sufficient information from a single image in order to create a three-dimensional model. These chapters discuss the background to the concept of single-image data capture and the development of a solution, the Phoenix approach.

Chapter 7 details the foundation of the techniques developed which lie in the mathematics of photogrammetry and artificial intelligence. Building on the photogrammetric techniques discussed in Chapter 6, and on research from other disciplines, new ideas and possible solutions are given. It is possible to interpret the shape of an object, by combining the mathematics of projection with knowledge of the

object. Areas currently being researched by the artificial intelligence community to allow simple machine vision are also shown as examples of fully automated approaches to the problem.

The Phoenix algorithms produced during the research for the solution of the single image problem, produced by the VHN project, are detailed in Chapter 8. The mathematical derivations of these algorithms are also given. Two techniques are illustrated, showing the development of the algorithm based on further research and the initial testing. The technique follows on closely from the graphical solution to the single image problem discussed in Chapter 7 as well as drawing on areas of photogrammetry.

Chapter 9 shows the results of tests made of the Phoenix algorithm. It compares the Phoenix technique to more standard approaches of photogrammetry and theodolite survey as described in Chapter 6. As a final illustration of the technique, a historic photograph is used to produce a three-dimensional model for inclusion in the VHN project. The model then follows the flow-line ideas discussed in Chapter 6 to produce an object in the VR software, showing a practical example of the model structure illustrated in Chapter 5.

The final chapter of the research aims to draw together the concepts described and developed throughout the research and present an overview of how these will aid GIS to develop. In doing this, it will illustrate the particular areas of the research where the aims and objectives have been answered, as well as areas where further research and development is required. As discussed in Chapter 2, current computing technology efficiency is progressing at a faster rate than ever before, leaving GIS behind. Part of

this research aim is to show that, through a combination of existing technologies and innovative research, GIS can take full advantage of this rate of change.

2. Evolution of Spatial Information

Geographic information has always been utilised by society, from early hand drawn maps, to the highly technical GIS of today. This information is used to show a representation of the real world in a presentable format, i.e. a map, in addition to being an interface to other information, such as databases and multimedia. Changes in the use of geographical information have been caused by the advent of the microprocessor, and in particular, the recent advances in affordable processing power. The awareness of digital data has led to an increase in geographic information's usage and availability, with data that was originally stored in a paper form being held digitally.

The influence of computer technology on geographic information has made new applications, such as VHN, imaginable as well as possible. This chapter outlines some of the current changes occurring to geographic information with relation to the VHN project. By looking at current research topics in GIS information, applications, and users, the chapter introduces aspects of multimedia and temporal data, the representation of three-dimensionality, and the application of VR and the Internet. These technologies are examined to see how they can work together to build up the elements of the VHN project. Here, the aim is to supply a web based interface, accessible via the Internet, which can interact with objects and components through defined spatial standards. These components will interrogate the underlying data stores that contain three-dimensional information captured by existing and new techniques such as the Phoenix approach discussed later, allows for the creation of multimedia virtual worlds for the user to explore.

2.1 Geographic Information Systems

Developed in 1963, one of the earliest geographic information systems, the Canada Geographic Information System (CGIS), provided research into many conceptual techniques still in use today. The purpose of CGIS was to analyse the data collected by the Canada Land Inventory and to aid in developing land management plans for rural Canada (NCGIA 1990a). Goodchild (1987) suggests that a GIS consist of four parts; input, storage, analysis, and output. CGIS and other prototype GIS applications in the 1960's pushed the limits of available computer technology for all of these parts to new levels. For example, the creation of CGIS saw the development of scanning, data storage, and plotting.

The rapid uptake of computer technology in the past few years has been aided by the cost of new computer hardware decreasing, while the power increases at a rapid rate. In 1965 Gordon Moore, co-founder of Intel, stated that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented; this has become known as Moore's Law (see Figure 2.1).

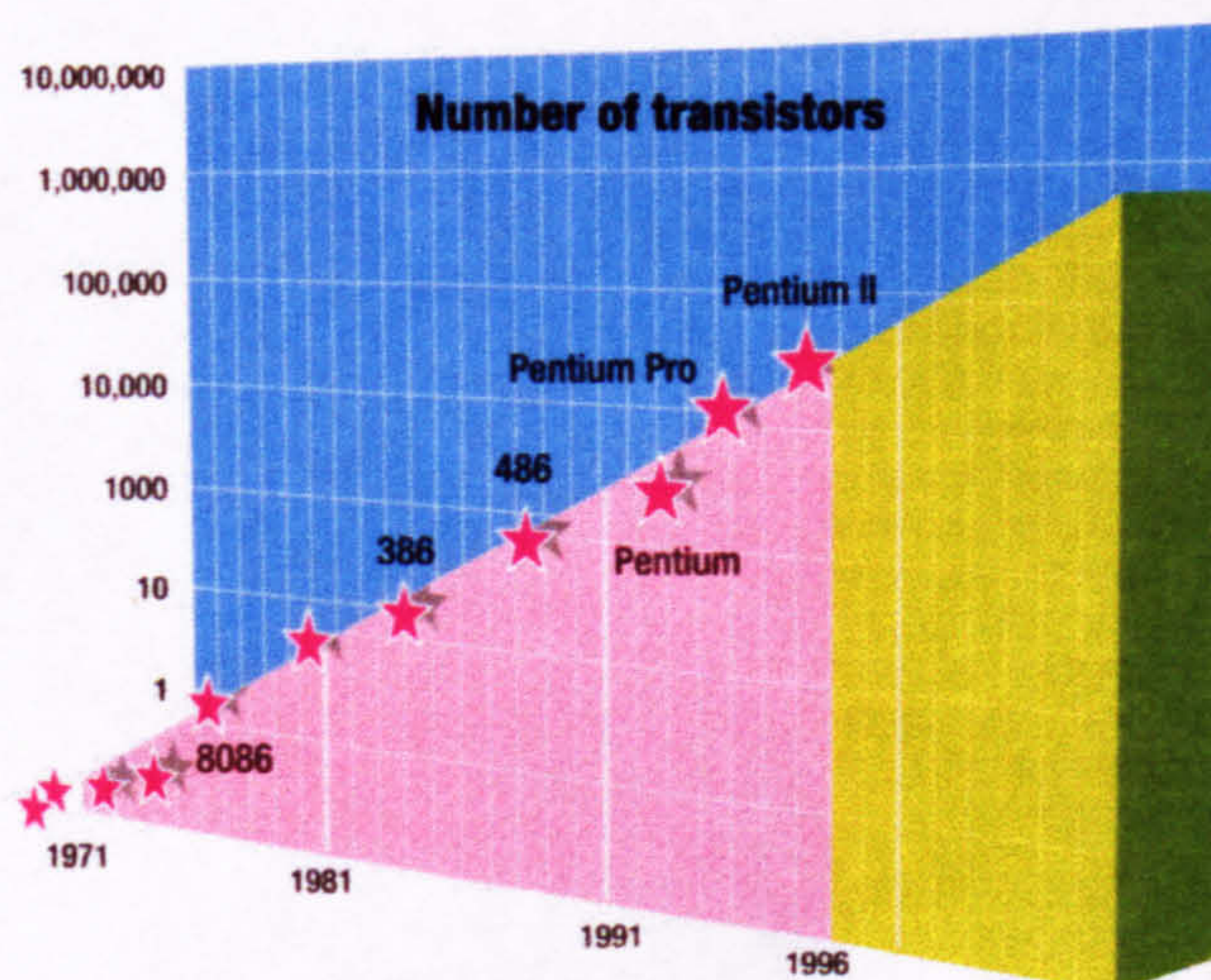


Figure 2.1: Moore's Law

Figure 2.1. Moore's Law predicts that the number of transistors that can be packed onto a silicon chip (shown on vertical axis) doubles every 18 months (Mullins 2000).

In subsequent years, the pace has slowed, but data density has doubled approximately every 18 months, this has become the current definition of Moore's Law. Most experts, including Moore himself, expect Moore's Law to hold for another two decades (ZDNet 1998). This growth in power is in all areas of computing, from the introduction of faster processor chips to larger storage devices.

In the past 37 years since CGIS was designed, computer visualisation techniques have developed. However, the cartographic roots of GIS visualisation are apparent, with digital maps mirroring their original paper style. The computers of today can manage all the dimensions available in the real world, length, width, height, and time, though the digital map cannot. Since the start of this research, commercial systems have begun to extend to enable three-dimensional data and allow access to distributed data. VR and Internet technologies are the main way in which GIS have moved forward, away from pure analysis. The 3D Analyst extension for ESRI's ArcView GIS and ERDAS' VirtualGIS module for Imagine give the user the ability to view data in three-dimensions. VirtualGIS and 3D Analyst allow for the interactive perspective viewing of two-and-a-half-dimensional data, i.e. positional data with height, such as a triangulated irregular network. Similarly, Autodesk's MapGuide and ESRI's Internet Map Server give accessibility to basic GIS functionality by way of the web.

The use of emerging technologies, for example, low-end VR, multimedia, computer gaming, and the Internet, will improve the visualisation of data in GIS. For this to happen it is necessary for all four parts of GIS, as identified at the beginning of this section by Goodchild, to be examined and enhanced. Although the technical aspect of GIS is being actively developed, the display capabilities are beginning to lag behind those found in other areas of computing. This is best summarised by the statement that,

“For thousands of years traditional mapping has used a variety of abstract symbols, colours and patterns to characterise a landscape. Conventional GIS software extended these procedures to the digital map and linked map features to attribute data. Advanced analytical tools provide entirely new ways to address complex spatial relationships and model alternative actions. Until recently, however, the rendering of GIS results primarily has been restricted to the same set of display techniques used in manual cartography” (Berry *et al.* 1998, page 47).

Taylor (1991) suggests the visualisation of data in a GIS can be thought of as a triangle with formalism, cognition, and communication forming the sides (see Figure 2.2). Without equal development for GIS to all three sides of this visualisation triangle, these systems will not be able to access the full power of today’s visualisation technology.

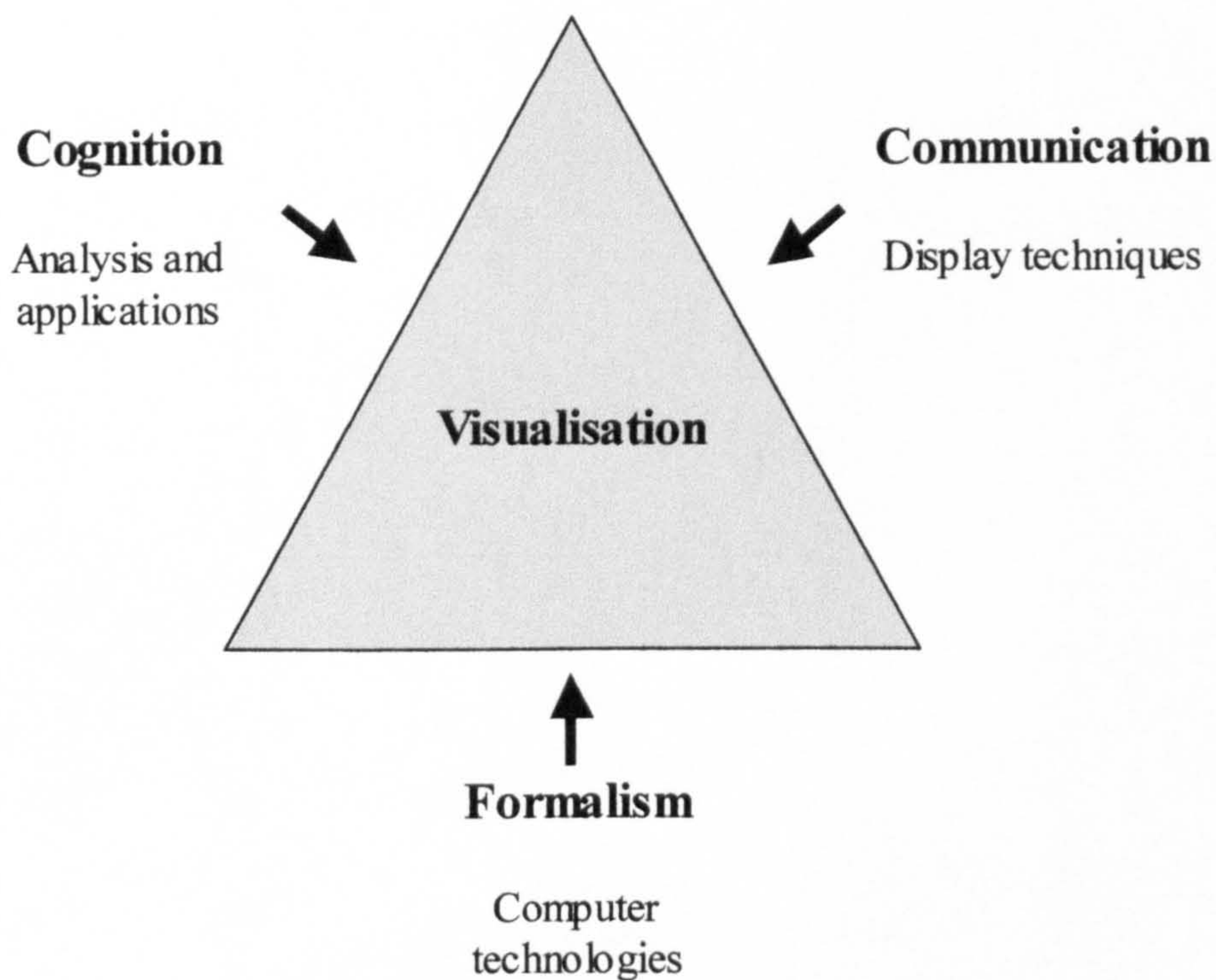


Figure 2.2: The components of visualisation

Figure 2.2. Visualisation in a spatial map is an interaction between formalism, cognition, and communication (Taylor 1991).

Current GIS research and development is concentrating on adding cognition to the visualisation triangle, focusing on the analysis and applications of the data. This has seen the area of spatial cognition remaining in a two-dimensional environment. In computing in general, the current development emphasis is on the sides of communication and formalism. These two areas are being highly developed towards three-dimensional data in the computing world, for example by hardware manufacturers, CAD, VR, games, and Internet developers. Although it is now possible to produce visually intensive three-dimensional images, GIS applications use these for little more than display.

Many of the problems and applications of spatial data are resolved by reducing the data to two dimensions. This has come about through data only being available in two dimensions, as well as education in spatial problem solving concentrating on two-dimensional data. Initial three-dimensional mapping, as produced by 3D Analyst and VirtualGIS, consists of extruded objects on a landscape or terrain. These objects do not have any real form or detail, such as doors or windows, except for the application of textures where possible. More detailed three-dimensional objects, such as those designed by architects for planning applications, tend to appear alone on a flat terrain.

This research suggests that Taylor's diagram has extended to two triangles, as shown in Figure 2.3. Cognition is being researched for two-dimensional data sets, whereas research into communication and formalism is being carried out with the goal of visualisation of two- and three-dimensional data sets both locally and remote to the data set. It is these ideas that are currently at the forefront of GIS marketing and research. The blank sides of the triangles in Figure 2.3 are supported by the existing technologies for cognition, communication, and formalism. Although existing GIS can perform

analysis on three-dimensional models, for example line-of-sight, they lack the ability to portray the information. The display options present are rigid and not user friendly to an investigative low-level user.

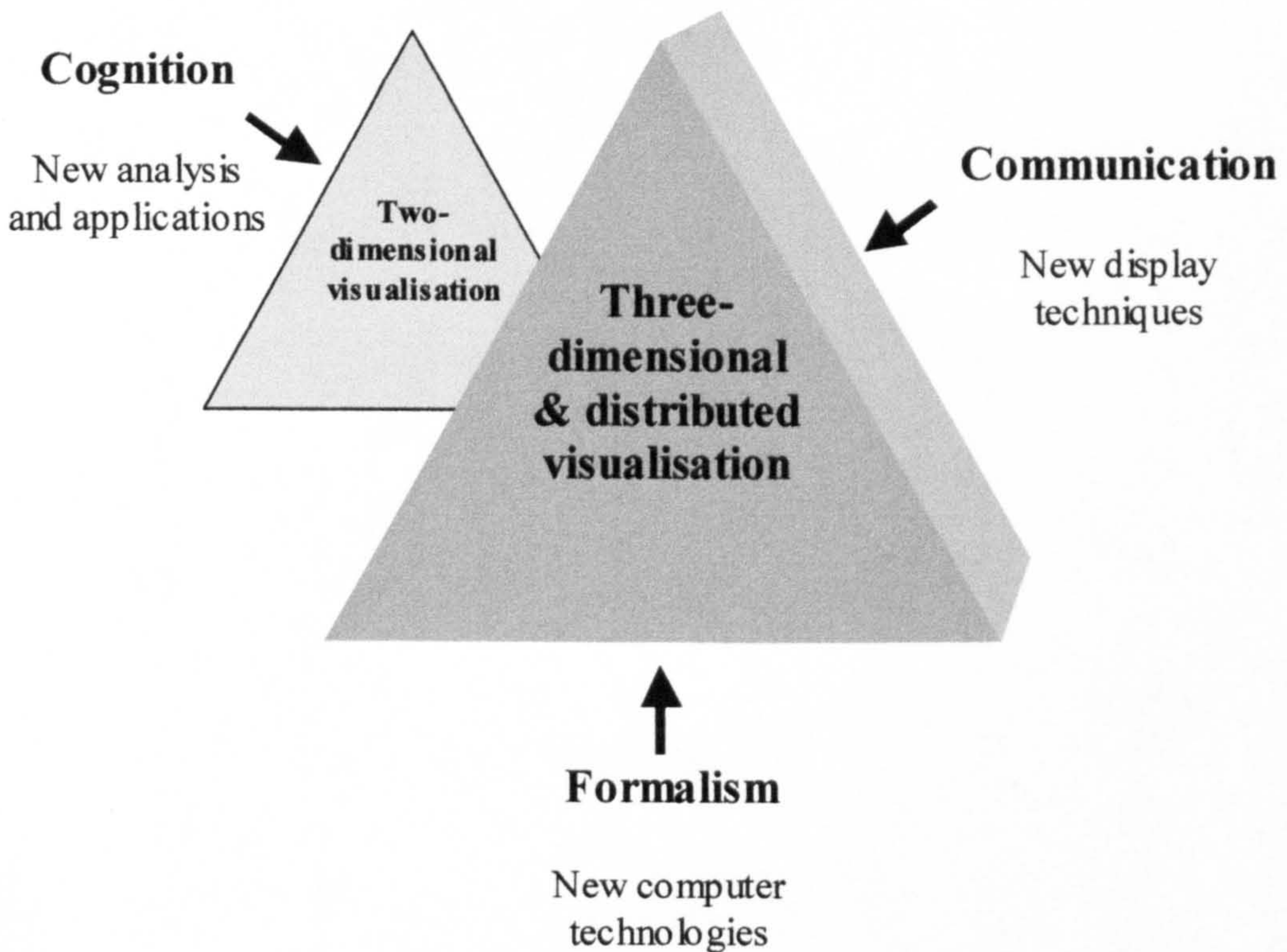


Figure 2.3: Taylor's components of visualisation adapted to current situation for GIS

Figure 2.3. The current situation for spatial information is that research is into the two- and three-dimensional aspect of communication and formalism, both being highly developed in the computer world. However, the cognition aspect is still concentrating solely on two-dimensional data.

Many of the barriers to the improvement of GIS exist due to the origins of the information in these systems. It is therefore necessary to address the problem at the level of the data placed into the GIS, rather than producing systems to allow the three-dimensional viewing and distribution of existing data. Currently, most data, and problems, are two-dimensional and remote and therefore there is little gain in the creation of a three-dimensional networked GIS. Applications such as the VHN project are changing this, and the research suggests that it is the information as well as the system that needs to develop to encompass fully the new technologies.

2.2 Geographic Information

The representation of reality exhibited by most maps incorporates the assumption that the map user is aware of the impossibility of displaying the three-dimensional nature of the environment using an essentially two-dimensional tool. Today, digital cartography still mainly uses paper-based techniques, although there is the power and technology to produce realistic three-dimensional models of surroundings (Fairbairn and Parsley 1997). Maps represent an orthographic projection of the three-dimensional world; the surveyed three-dimensional data being converted to two-dimensions when stored. With the digital storage of geographic information, the concept of the map, as such, is no longer required. The conversion, from three-dimensional data to a two-dimensional view, should be done at the time of viewing and not storage, avoiding any loss of data and allowing for future viewing from any direction.

A major data supplier of geographic information to the British market is the Ordnance Survey (OS) of Great Britain. The OS completed the conversion of their large-scale geographic information stored on paper maps to a digital form in 1995. For the purpose of this thesis, the term large-scale is used to refer to data captured for scales of 1:2,500 and above. This partially covers the OS Landline range of data, as Landline also includes 1:10,000. This data is inherently two-dimensional, using a Cartesian (easting and northing, or x and y) co-ordinate system. Some height information is stored as textual attributes, for example spot heights and bench marks, with the three-dimensionality of the landscape available separately in the Land-Form PROFILE product, as either contours or digital terrain models (Rhind 1995). However, no information is available on the height of buildings or other features on the landscape. A rival data supplier, GeoInformation International, has produced digital city models from large-scale aerial photography. Their product, City Heights, is a vector data set

comprising building footprint, building height, road centreline, and road centreline height. This allows for the more accurate inclusion of these objects into a three-dimensional data set. Currently, City Heights is only accurate to within 0.5 metres and not all of the country is available (GeoInformation International 1998). Chapter 6 discusses further means for the capture of three-dimensional information and the related accuracy and uses for the resulting data.

The introduction of competition from new companies in the market place has increased the amount and variety of spatial data available to users. Digital geographic data, for example, the OSCAR road network and Strategi data from the OS, is moving away from being a representation of paper mapping to becoming data suitable for usage in GIS. This has allowed the less technical user access to data that does not require as much formatting and structuring before use, opening up new sections of users in the community.

The VHN project requires large-scale information for the creation of the base model and the positioning of objects in the virtual world. Section 6.1 shows how existing two-dimensional data, such as the OS Landline product, can be converted to a simple three-dimensional model. This conversion is further simplified and enhanced through the inclusion of more data sets, such as City Heights and Strategi.

A reason for the increasing demand to push geographic information forward technologically is the increased user base. Digital maps are no longer purely for the academic, the researcher, or utility companies. A more common user today is the tourist in the street looking for a hotel, or a driver searching for a route round a traffic jam. These people see the world in a different way to previous digital map users. Davies (1994) suggests there is often a conceptual gap between the users and the

system's understanding of what is going on. With the addition of more low-level users, this gap is liable to increase. Hence, a GIS is required that is as intuitive to use as some contemporary applications, such as spreadsheets and word processors.

The range of methods of map production is expanding, it is now possible to create spatial representations from spreadsheets, statistical packages and GIS, as well as from CAD tools and illustration software. The former applications are data-driven methods in which the map creator (who is also very often the map user) has limited control over the graphic output. The latter are user-driven methods, which give considerable choice over representation methods and style (Fairbairn and Parsley 1997).

Ghormley (1995, page 52) states that the problem with GIS applications is

"...that they're created by professionals for professionals"

and suggests that the next-generation of GIS should have

"...easier data collection, better preparation and integration tools, and much simpler ways to combine and present project results" (Ghormley 1995, page 50).

In the five years, since this was written, little has been done to make the process of populating and using a GIS easier for the user. Increased automation has made the area of data collection easier and more digital data is available pre-prepared than then. However, the integration of this data with existing data and the presentation of this data have remained static. The next two sections discuss new data formats that users are becoming accustomed to, through the use of other applications and computer games. The sections suggest how these may be made available in GIS to aid the user in the both analysis and presentation of results.

2.3 Multimedia and Temporal Data

Raper (1994) states that developments in computer technology have made a fully-fledged multi-dimensional multimedia GIS a possibility for the masses: including the public. Since this statement, computer technology has developed at increasing speed, yet in 2000, GIS software mainly works with numerical data and structured text. Standard relational database models allow for the storage of formatted text, integers, decimals, dates, and times in an alphanumeric form.

For implementation into a GIS, the position of multimedia data in the overall conceptual model has to be examined. Lombardo and Kemp (1997) suggest that the position of multimedia is governed by

- whether the multimedia object represents an entity or an attribute of an entity aggregation
- whether the entity or attribute represents a spatial or an aspatial property in the overall schema
- whether the application requires the multimedia data to be merely displayed (or reproduced in the case of audio objects) or to participate in content-based retrieval, in predicate-query mode or browsing mode

Multimedia is concerned with storing binary data, for example, images, unformatted text, audio and video, as well as the more traditional data types (Kemp 1995). The data can now also be time dependent in the case of animation and audio. The interaction of these data sets when presented to the user, for example the synchronisation of video and audio clips, is also an aspect of multimedia. Some database packages, such as Visual dBase, already allow for storage of images and audio. Rather than actually storing the audio or image inside the database, the system maintains a reference, for example a filename or alias. This means it is not possible to search the multimedia data by more

than use of the filename. GIS users require the ability for the multimedia data to be both aspatial and spatial, and to allow the possibility of in-depth searches. An example search could be 'Show me the location of all birds with a song similar to that of a wren'. In this way, it is necessary to include multimedia data, not as a separate data type, but as an integral part of a GIS data set.

Technology for more intuitive searching of multimedia exists, but has not been implemented into a commercial GIS application. Virage produce image-matching technology that has been implemented into the AltaVista Search Engine (<http://www.altavista.co.uk>). This software allows a user to enter textual search criteria in order to define a query to locate images associated with a keyword on the web. Once an image has been found, the user can refine their search by either adding to the criteria or searching for visually similar pictures.

Figure 2.4 shows the results of a search for images of the mythical bird, the Phoenix. Although the majority of the returned images illustrate the expected answer, anomalies exist due to how the images are catalogued. The application, which automatically traverses web pages to create the database, uses the filenames of the images, any associated text that forms a title for the image, and keywords that appear on the same web page as the image. The example search of '+phoenix +bird', can be conceived by the user as returning an incorrect result where the two words phoenix and bird are found on the same web page.



Figure 2.4: Images found when searching for +phoenix +bird

Figure 2.4. AltaVista Photo Finder searches for images based on keywords or visual prototypes. The images above are those found by the software for the search +phoenix +bird. To explain the incorrect images, (a) is of the ‘Phoenix Bird’ air race team and (g), (h) and (j) are caused by the words phoenix and bird appearing on the same web page.

The search can also find visually similar pictures, by comparing the distribution of colours in the image. This can have far more sources of error than a textual search of keywords related to the image, and is shown in Figure 2.5. Errors are caused mainly through images having a high concentration of colour in part of the image. In the illustrated case, the high concentration of blue in the lower part of image (a) has caused this to feature prominently in the search results.

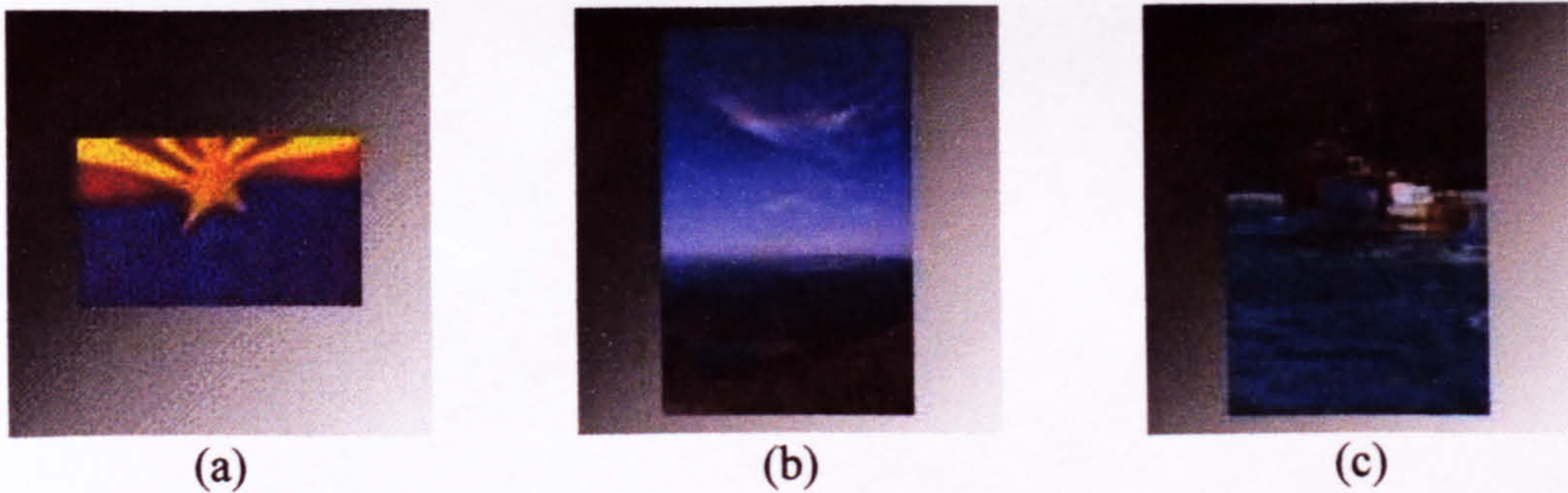


Figure 2.5: Example of visually similar images

Figure 2.5. Images (b) and (c) are examples of images found when searching for images visually similar to (a). It can be seen that the high concentration of blue in the lower part of (a) has influenced the result.

Presently in the AltaVista interface there is no ability for users to sketch, or supply, their own image and have visually similar pictures found. Applications of this technology to the analysis of spatial information include the ability to search for network patterns and urban developments, for example linear towns or concentric rings.

Virage's Video Cataloguer software allows searches to be made on video data sets. This works by splitting the video up into segments based on changes in the visual content of the frames. Sets of distinct frames are extracted from these segments in order to create a storyboard interface to the user. Similarly, the software can search the audio of the video to catalogue words or sounds (Virage 1998). In this way libraries of videos can be stored and then quickly searched for either similar scenes or sounds to the criteria specified by the user. This software is also demonstrated at the AltaVista site.

Applications of a multimedia GIS exist already in the marketplace with the attachment of multimedia data, as an attribute of geographic data, becoming more common. What is required from the market place is the motivation to turn multimedia data into an integral part of a GIS rather than just an attribute. This will make new data sets available for analysis, enabling the user to increase their understanding and perception

of existing data, for example, through the automatic creation of three-dimensional models from photography and video by use of imaging techniques (Chapter 6). By employing multimedia data, for example sounds, in a virtual world such as that created for VHN, a greater sense of atmosphere can be created for the archaeologists using the data. However, by allowing the archaeologists to perform queries on this data, for example 'where can the church bells be heard from' will help to bring a greater understanding to the spatial structure of early settlements.

2.4 Three Dimensionality

Digital mapping is reminiscent of the world described by Abbot (1884) in the book, *Flatland*, rather than the spherical Earth promoted by Aristotle and Pythagoras. To date, three-dimensional visualisations have been restricted in their use. The main areas tend to be in the representation of subsurface geo-objects, visualisation of spatial heterogeneity estimated by simulations, or pre- and post-processing of data for numerical models (Turner 1995).

Various methods have been developed to display the third dimension on paper, as illustrated in Figure 2.6. Originally, sufficient data was not available for detailed maps of land surfaces and so these were represented pictorially. As the data accumulated, verbal descriptions of surface phenomena evolved to graphic symbols, representations of slope, and finally, and with the availability of survey data, to the mapping of elevation by contour lines. More recently, shaded relief has been used to assist visualisation of land surfaces in situations where contour lines alone are inadequate (McCleary and Jenks 1993).



Figure 2.6: The evolution of the mapping of land surface form

Figure 2.6. Originally sufficient data was not available for mapping of land surfaces and relief was represented pictorially. Slowly graphic map symbols evolved to give representations of slope and with accurate surveying the mapping of elevation by contour lines (McCleary and Jenks 1993).

Current techniques associated with computer graphics, such as geometric transformations, depth cueing, texture mapping, shading and colouring, can also be applied to cartographic data (Kraak 1993). McLaren and Kennie (1989) categorise the various methods of display of three-dimensions by the variations in symbolic content and degree of realism, shown in Figure 2.7.

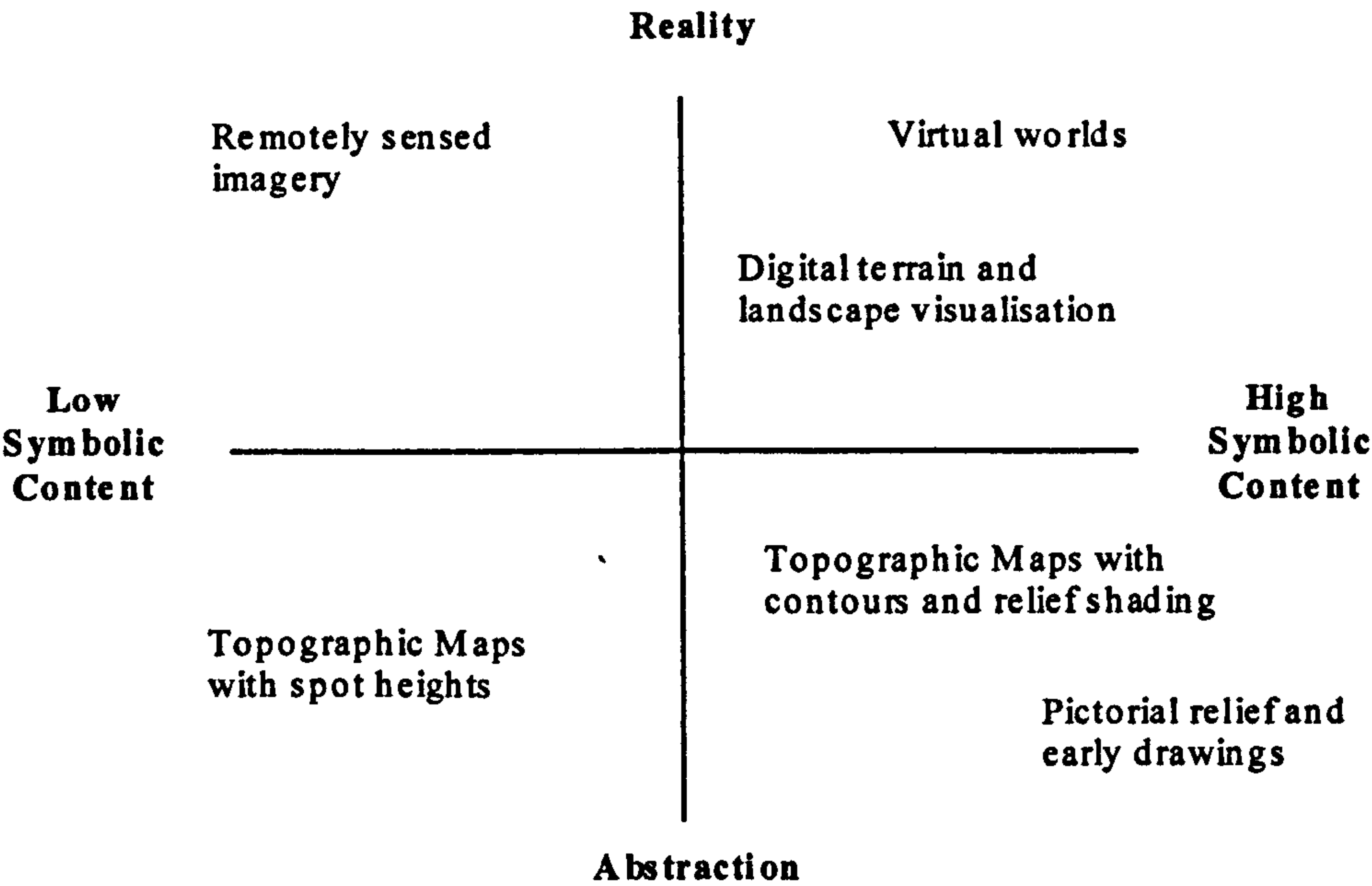


Figure 2.7: Classification of techniques for visualising terrain

Figure 2.7. The representation of relief is a fundamental component of the cartographic process. A wide range of techniques has been developed and these vary in both their symbolic content and in their degree of realism (based upon McLaren and Kennie 1989).

Most current GIS applications display three-dimensional data as topographic maps with spot heights. More advanced systems allow for the creation of perspective views of the data, often referred to as two-and-a-half-dimensions.

Figure 2.7 suggests that the virtual world approach produces a high degree of realism and has a high symbolic content (McLaren and Kennie 1989). This research examines the possibilities for a more realistic approach of digital terrain and landscape visualisation than that produced from the usage of virtual worlds. The cartographic design aspect of GIS has become stunted by its own success, in being able to portray the third dimension using only two. Collinson (1997, page 117) states,

“We are now at a point in the history of cartographic design and production where it can supersede all that has gone before, so why is it not doing so?”

This statement is still true three years later. Consequently, GIS has remained in its two-and-a-half-dimensional world. With more data being used digitally, the higher dimensions can be used. These maps can have reflections, shadows, horizons, and sunsets over them (Figure 2.8). A further improvement is the inclusion of multimedia style animations to illustrate new data sets and higher dimensions, such as time.

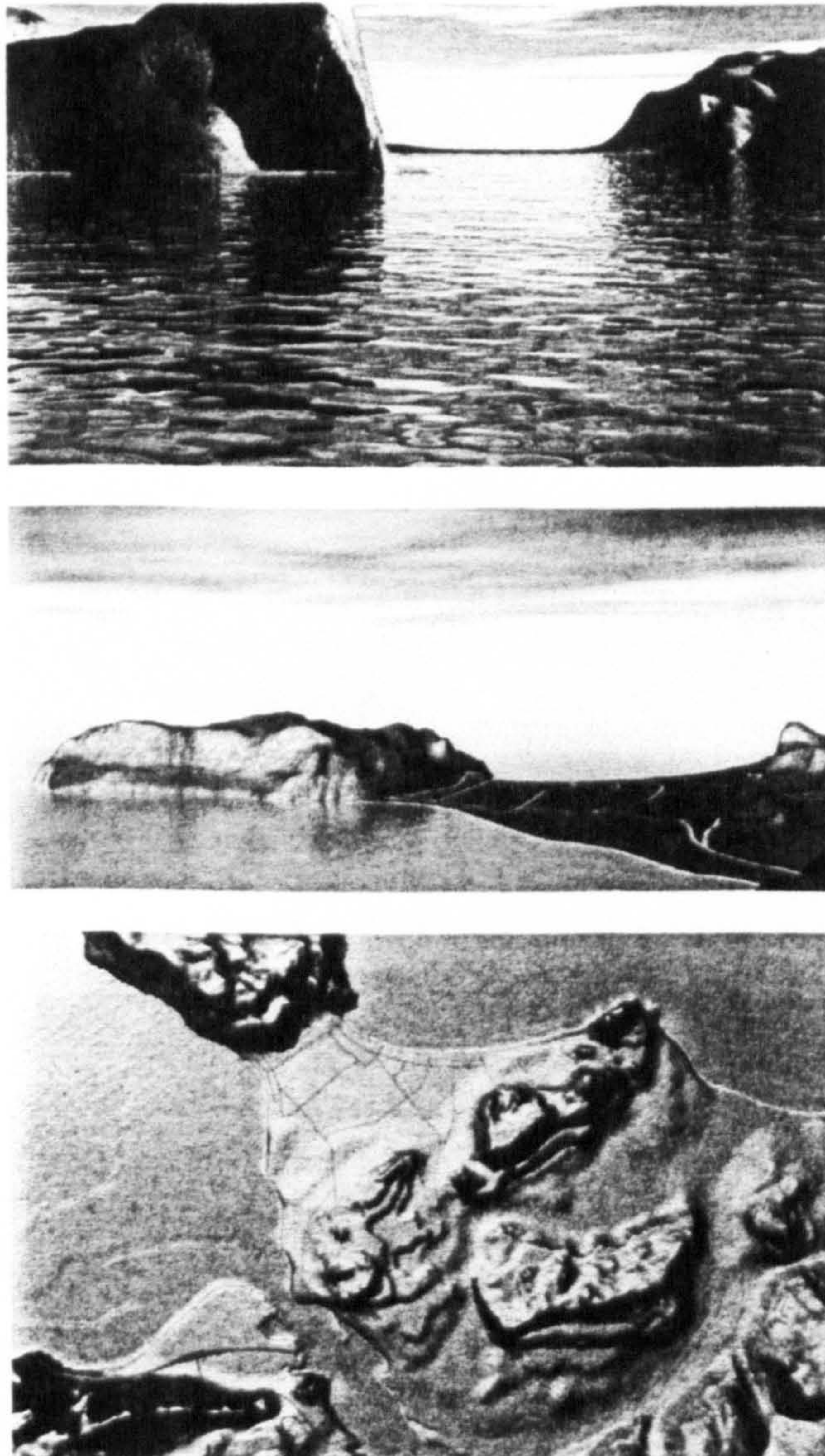


Figure 2.8: Virtual worlds created from accurate mapping data

Figure 2.8. The conversion of digital mapping data to three dimensions means that maps can now have, shadows, seas, waves and horizons, while viewed from any angle (Collinson 1997).

These techniques, when used to their full potential, can construct virtual realities difficult to differentiate from the real world, seen to good effect by their increased usage in the film industry. An advantage of having three-dimensional geographic information is that the user can interact with a world that begins to look and behave more realistic. With a three-dimensional data set, improved visibility algorithms can take account of buildings along with landscape; pollution models can now have the effects of wind tunnelling down streets added to the model. These are just two examples, although others exist. In current data sets this added three-dimensional knowledge to the model no longer exists through removal at the data capture stage, or is not to a high accuracy.

While it is useful to have near realistic models for display in some cases, for example presentations, applications such as VHN are better off with more simplistic models that avoid overloading the user with information. This does not mean reverting to blocks of solid colour, but using what may be considered a half way house of simple textures and basic shadows to aid the user in their perception of the objects and their depth. The use of three-dimensional information means that GIS have to alter the means used to display the data. A top down plan view of the information as supplied by current GIS will mean that the user is unable to see some of the information being presented to them. The next section discusses the topic of virtual reality and the associated display techniques used to give the user an enhanced feeling of the space of the data.

2.5 Virtual Reality

Different from traditional three-dimensional animation, VR software calculates, generates, then displays, the current true perspective view of the virtual world many times a second. Fairbairn and Parsley (1997) explain that ordinary animations rely on the displayed views being calculated and generated some time previously and then

stored in order. Here, no interaction is possible with the animation as the viewer can only passively watch the pre-created sequence, whereas, in VR a world can be explored and examined continuously from any perspective in real time, at will. The application can impose physical characteristics on the world and the user, for example gravity, or allow the user to float above or below the virtual world to gain a different perspective on the information being displayed. Being in continuous control of the movement and viewpoint is an experience that makes the virtual world appear to become almost real, hence the term virtual reality.

Hacklay (1998) outlines the ideal VR GIS as described by Faust (1993) as containing

- very realistic representation of real geographic areas
- free movement of the user within and outside the selected geographic terrain
- standard GIS capabilities (query, select, spatial analysis, etc.) in a three-dimensional database
- the visibility function should be a natural and integral part of the user interface

Currently GIS applications are trading off these four ideals against each other, for example, 3D Analyst allows queries to be carried out in the three-dimensional environment, but does not have a realistic representation of areas. Buildings, for example, can only be modelled as extruded polygons. Conversely, ERDAS Imagine is developed to allow the realistic representation and free movement over a data set. However, this is at the expense of being able to query and analyse the model through the three-dimensional interface.

The VHN project initially approaches these ideals by choosing to leave the very realistic representation of data until there is the computer power to handle the graphics and data.

Where the approach to modelling the virtual world for VHN differs from most existing models is in the use of detailed three-dimensional buildings on a detailed three-dimensional landscape. Large-scale townscapes that have previously been produced have a tendency to be created on flat planes. Chapter 5 details further information on the required structure of the data to allow for this type of model.

Gillings and Goodrick (1996) proposed a change to how VR was used with real world information. They suggest that VR should not be used as a method for imitating reality, but rather as a tool to simulate aspects of the real world. Commonly referred to as virtual worlds, these create a synthetic environment for the user to examine and explore. These worlds are commonly thought of along with the idea of VR, though it is not a necessity for a virtual world to be viewed by VR techniques. It is important to realise that VR is not the model but a tool with which to view that model. The research here is concerned with the methodologies for the design and structure of the model - though here it is of the real world rather than an imagined world.

VR enables a move towards a finer emulation of the real world, which can be combined with our abilities to navigate through environments. Clearly, hardware considerations are important in the successful implementation of VR systems and can determine the success of a product. The provision of complex and rapidly changing images in flight simulators, for example, requires powerful storage, computational and display technologies. The raising of the profile of the participant from a passive recipient of data to an active initiator of system behaviour, means that the human-computer interface is as important an issue as computer power in VR (Fairbairn and Parsley 1997). Chapter 5 details more information on these data structures and concepts.

VR systems vary in these hardware and interface capabilities. There are generally considered four levels of VR, immersive, transparent, projection, and desktop. Full, or immersive, VR requires the participant to be subject to stimuli affecting many senses, including vision, hearing, balance and touch. Such systems require head mounted displays, audio speakers, moving platforms and tactile gloves, and are expensive and currently, unwieldy. They are applied to completely artificially constructed environments. A prototype system, from VR Systems UK, proposes encasing the participant in a translucent sphere onto which the virtual world is projected. This enables the user to walk naturally, run, jump, or crawl in any direction (Figure 2.9).

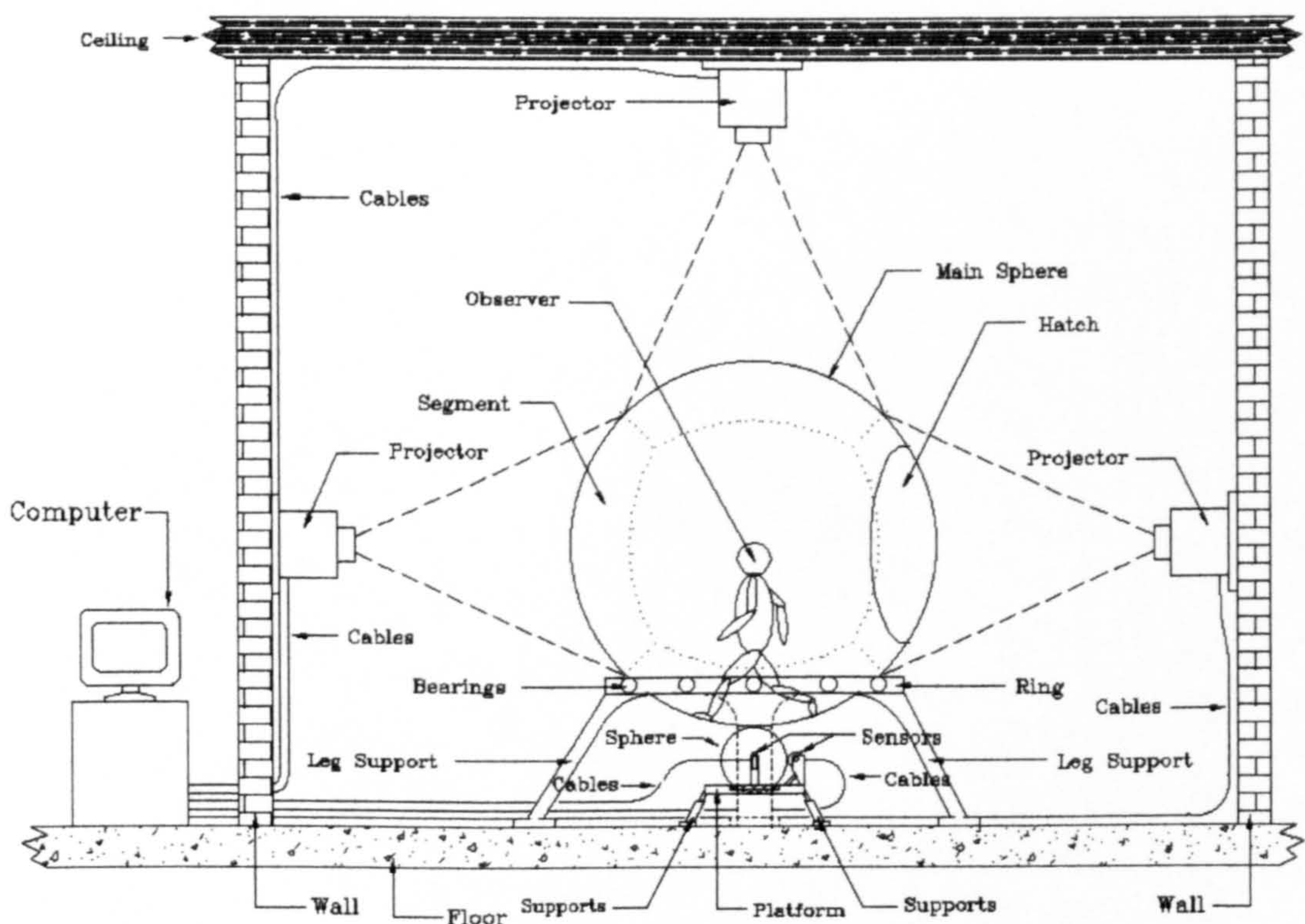


Figure 2.9: The Cybersphere - a fully immersive spherical projection system

Figure 2.9. The Cybersphere enables the user to walk, run, crawl, or leap in any direction inside a virtual world. This removes the restrictions of current head mounted display systems (Eyre 1998).

The Cybersphere system allows the user to interact with the virtual world in a more natural way. The display from the five projectors located around the sphere is updated

in response to the movements of the main sphere, controlled by the user. The resulting movement exerted on the smaller sphere and relayed back to the display computer interprets these movements. This system is still a prototype, although a final production system is expected within two years (Eyre 1998).

It is hoped that the Cybersphere approach may remove some of the health factors associated with current fully immersive systems. Many users of head mounted displays often complain of nausea after prolonged use. This can be caused by the mathematical model of perspective not being exact and the slight time lag between movement of the body and update of display (Seymour 1996). Presently, the discomfort and the clumsiness of the equipment used to display the imagery is a restriction to the usage of fully immersive VR.

Transparent VR, uses the 'real world' as a backdrop, seen 'through' the device presenting the spatial information. The most common form of this device is to display the image on a visor in front of the user's eyes, similar to an aircraft pilot's heads-up-display. A development in this field is the virtual retinal display (VRD) designed at the Human Interface Technology Lab (HITL). Three photon sources are used to generate coherent beams of light to draw a diffraction-limited spot directly on the retina. The beam of light is modulated to match the intensity of the image being rendered. In a similar way to how a television picture is made up, the beams scan across the surface of the retina building up the image. The system allows for transparent images or full images to be built up (Tidwell 1996).

Unlike such individual interactions, Projection VR can be a multi-participant experience involving the presentation of large-scale graphical displays, for example in a planetarium-like situation. This is best used for situations where a three-dimensional

impression of an object is required to be viewed simultaneously by a group of people. It is common for a single user, often at the rear of the room, to be in control of the motion around the three-dimensional world. Unlike traditional situations where a simple animated movie of a model may be displayed, the VR capabilities allow those viewing the model to interact with and move around the model at will.

Desktop VR is the most commonly used form of VR system, because it can be presented on standard computer monitors. This form of VR is the one used for this research, with various software applications used to create and view artificial worlds on the desktop and over the Internet. The downfall with common desktop applications of VR is the lower processing power of the systems displaying them. Often only simple models can be displayed if the interaction with the model is to be kept at a maximum. Increased processing power and dedicated three-dimensional graphics cards are aiding the situation, but many of the capabilities of VR are missing on these desktop systems.

VHN looks to use desktop VR, as this supplies the cheapest interface to the user that allows the full 'six degrees of freedom' of movement across the environment (that is three directional motions and three rotational motions). The project, however, is more concerned with the development of the data set rather than the final application that is used to display it. Therefore, with the correct generation of the virtual world, there is no reason why the data can not be displayed in any one of the above systems.

When applied to GIS, VR techniques, along with cartographic visualisation tools in general, encompass freedom of movement with realistic display of data. Display can be initiated and up-dated with reference to the data, but there is also considerable user choice in viewing position, scale of display, navigation through the data set and rendering of three-dimensional cues (MacEachren and Taylor 1994).

Fairbairn and Parsley (1997) state that VR is a non-deterministic method of presenting spatial information in graphical form. In this sense, it

- permits free and random access to data, giving a 'user-navigation' capability
- allows for dynamic exploration of a data set accomplished under interactive control
- involves the modification of the data whilst being viewed from a static position, or the modification of the viewing position whilst navigating through a constant data-set, or a combination of both
- creates an environment for the making of assumptions and testing of hypotheses
- enables the creation of user-defined views and maps
- (by extension) reduces reliance on a professional cartographer to create a map product

In VR, the characteristics of visualisation in cartography are extended using a number of further attributes. These are the extension to three- and four- dimensions, creation of artificial constructs such as gravity-free and barrier-free movement and viewing, and possibilities of complete 'immersion' in a spatial environment.

There are several fields in computer graphics that the mapping community has left untouched over the last years. Aspects such as ray-tracing algorithms can supply three-dimensional intersection algorithms. The games industry can also assist in the design of fast display algorithms. The area of file formats has also been addressed by others, the virtual reality modelling language (VRML, pronounced *vermel*) allows for the storage of three-dimensional data, attributes, and intelligence (VRML is dealt with further in Chapter 3). This research examines how these existing technologies can be encompassed into GIS, rather than reinvented.

Thoen (1995), Hacklay (1998), and CASA (1998) show that the trend, of research continuing to be carried out world-wide into incorporating VR and the web into GIS is developing. As mentioned earlier, systems such as ArcView, AutoCAD World, and ERDAS already contain the beginnings of these technologies. It remains to be shown whether these new commercial systems will benefit from being developed on top of existing GIS packages, or whether it is best to design new software. Figure 2.10 shows the range of projects investigating the use of VR for spatial data. The main areas of research are in the fields of urban planing, the environment, and data visualisation. These are all areas where realistic visualisation of a data set can aid in the interpretation of data. Little research is being carried out into making VR an integral part of GIS. The current trend is in preferring to export the data from GIS into separate VR packages.

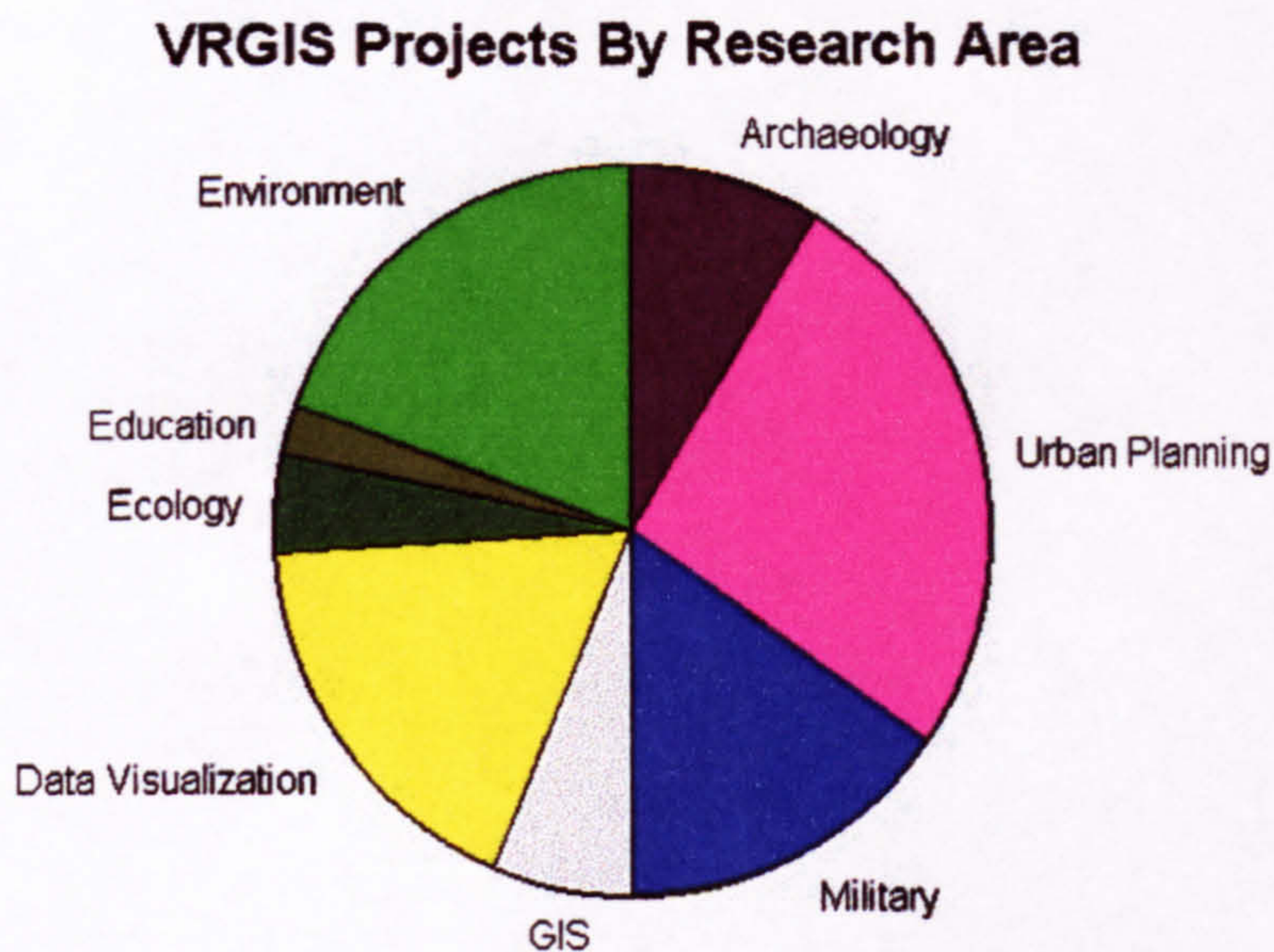


Figure 2.10: GIS projects using virtual reality

Figure 2.10. Recent research into the use of VR with spatial information has mainly been in areas involving visualisation (Hacklay 1998).

VR GIS may develop in one of two ways; one might be a system that can display three-dimensional images in a separate window, similar to how two-dimensional mapping is currently displayed. An alternative approach is as a complete three-dimensional

interface to an application. This method would involve the creation of a virtual GIS 'office' as the user interface. Here, all information from the GIS would be displayed in a three-dimensional environment. This would include the user interface as well as results and displays of other two-dimensional data. The approach of this three-dimensional interface lends itself more into allowing use of full immersive VR. An example of this type of application is the 'Virtual GIS Room' developed by the Environmental Systems Analysis Group at the New University of Lisbon (Neves *et al.* 1998).

Use of the three-dimensional GIS office may aid users in their conversion from both current paper based mapping systems and existing two-dimensional GIS. It would essentially be possible to recreate the users current view of their data in the three-dimensional environment. For example, plans could still exist in a virtual map cupboard, which the user can access, or a virtual computer could allow an interface to digital mapping.

2.6 *The Internet and The World Wide Web*

Increasingly, users are no longer just using one computer on a desktop in their office. The majority of companies today have their own connections to Intranets, Extranets, and the Internet. These are terms for how one computer is attached to another; the difference is in how far geographically it can access. Current GIS applications only tend to see data on an Intranet, where the networked location is mapped to a drive, forming a virtual hard drive on their system. These applications can use data from the Internet, as long as it is previously downloaded to a local hard drive. For many existing applications of GIS, this restriction is not disruptive. More regularly updated data is appearing on the Internet, for example, GPS positions, and satellite images. What is required is a system that can access data on the Internet as seamlessly as software can

currently access a file on an Intranet. Chapter 4 explains in more detail some of the current technologies and standards being developed for global access to data sets. Here two areas must be addressed, firstly the format of the data that is accessed, and secondly the protocol used to access the data.

Various methods of inclusion of networking ideas into GIS can be made. It is possible to connect to a machine running actual GIS software, for example ArcInfo. Alternatively, a GIS can be produced on the web, for example in Java, or GIS applications can include the ability to access files on the web. Each of these has technological aspects that need to be addressed. Today it looks increasingly as if web formats are the means for accessing and serving data over the Internet. However, current web technologies mean that it is not feasible to produce an entire GIS in Java or another Internet based language. However, there is the ability to produce ones that understand how to resolve and access files stored at locations on the Internet.

Networking spatial data and applications does not only allow for the visualisation and analysis of the data but also for a change in work pattern. These technologies also supply the ability for multiple users to interact with the data and other users by a combination of text, images, sound, virtual worlds and video. In this way, distant parties can participate in collaborative projects in real time around the world.

Chapter 3 will look towards web techniques, such as VRML and the dynamic hypertext markup language (DHTML), to give a combined approach, three-dimensional objects on a three-dimensional terrain. These techniques also give the possibility of distributed data and applications due to their basis on web technology. The new GIS will require the ability to communicate data to the user as well as allowing interrogation of databases and manipulation of parameters.

2.7 Summary

This overview shows what is possible for geographic information using today's technologies. Through these approaches geographic information moves away from being just a visual display of the world as it was with cartography. Instead of the two-dimensional cartographic maps of yesterday, it will be possible to have fully immersive three-dimensional worlds to explore and analyse.

GIS needs to develop so that visualisation becomes part of the whole process of GIS analysis and design instead of a by-product. Today's computing power has reached a level that allows geographic information to form an interface into other data sets and worlds. Unfortunately, many of the applications of the data that exist today restrict geographic information to two dimensions. The GIS of tomorrow needs to allow for the storage and analysis of three-dimensional data both held locally and remotely over networks.

It is impossible to predict where computing will have taken geographic information in five years time due to the speed at which technology is progressing. The boundaries of current hardware and software are being pushed and exceeded all the time with faster and more powerful versions being released and developed. However, GIS seems to be sitting back and waiting for these technologies to develop before including them. The way that computer technology moves now it is necessary to be incorporated at the beginning, or the systems will become left too far behind. This can already be seen with the level of three-dimensional interaction in CAD and computer games far exceeding that in the high-end GIS applications.

For the VHN project, it is necessary to pull together the diverse range of technologies summarised in this chapter to produce an interface to a database of historic information.

Developments in the web allow for the creation of a multimedia interface to the Internet and the related geographic information stored within. Using local and remote software components, the user is able to interrogate and display the data without duplication. This allows for collaborative working between remote groups around the world, and for the sharing of discrete data sets. As with any GIS, these data are the most important and costly part of the project. By extending existing two-dimensional data, and through new three-dimensional data capture techniques, more information becomes available for use.

The following chapters describe the technologies required for the VHN project and expand on those areas discussed in this overview. Starting from the web-based interface that the user will initially be faced with, the research moves on to the Internet and components that allow for the interrogation of the underlying databases. Options for the structure of the data that is stored are described, before more closely examining the requirement for three-dimensional data capture and in particular the solution to the problem of extracting information from a single oblique image.

3. Applying Web Technology

The ideas behind the web were conceptualised at CERN* in 1991, exploded with the Mosaic browser in 1993, and today are one of the main areas of software technology development. In examining the application of the web to geographic information, it is important to separate correctly the web from the Internet. Mitchell and Pendlebury (1996, page 67) state that

“The Internet provides an easily accessible mechanism for distribution of the information and the WWW [web] provides a multimedia means of displaying it.”

Mitchell and Pendlebury make the distinction between the web, which is about displaying information (usually in a graphical form) and the Internet, which is concerned with the storage, retrieval, and distribution of this information to web-based clients. The current monolithic approach to GIS, where one application does everything, now has to change to come in line with new web and Internet technologies. The introduction of the web and the Internet sees a shift to a situation where multiple interfaces, applications, and data stores exist, possibly in different parts of the world.

This chapter discusses the markup languages that exist to allow the display of information and applications that have been retrieved from the Internet. Chapter 4 follows on from this to discuss further the underlying technologies of the web. Beginning with the Internet, and continuing with object and component based solutions to allow for the transfer, or remote access, of applications.

* CERN – Conseil Européen pour le Recherche Nucléaire (European Laboratory for Particle Physics) found at <http://www.cern.ch/> on the web.

The VHN web page (Figure 3.1) forms the interface between the user, the components embedded into it, and the underlying data, allowing all three to communicate and interact with each other. For example, the user clicks on the two-dimensional map to change the position of the three-dimensional view, and a search for a related textual description on the area is executed and the results displayed. By allowing this interaction and maintaining the separation between application and interface, the functionality can be as simple, or advanced, as the developer requires. The interface is created using files that contain the markup and style with which to display the information. The following sections discuss the format of these languages and the elements of the interface they affect.



Figure 3.1: Screenshot of Virtual Historic Newcastle prototype

Figure 3.1. Virtual Historic Newcastle uses various aspects of web-based data and application distribution to produce an interface to spatial data.

3.1 Hypertext Markup Language

The Hypertext Markup Language (HTML) is the basic format used for storing textual information that is to be displayed as 'pages' on the web. Originally developed by Tim Berners-Lee in the late 1980's, HTML derives from the Standard Generalised Markup Language (SGML). By the use of a web browser, users can access remote documents from web servers (using the http protocol) and interpret them for display on a client computer. Responsibility for the publication of standards for HTML and other web related formats resides with The World Wide Web Consortium (W3C), however these are often months behind the technology used in the browser software.

The more recent developments of HTML occurred during the 'browser war' between the two main browser manufacturers of the late 1990's, Netscape and Microsoft, and saw a rapid development in the features of HTML in a very short time. This led to a lot of style information being included in the markup for the text, enabling it to be displayed in various fonts and colours. Through this the language moved away from some of the W3C's original aim, which was to keep the style and formatting of a document separate from the information it contained.

A basic HTML formatted file is a text file that contains specified tags to suggest to an application (most commonly a web browser) how to layout the information within the document to the user. HTML is designed for the encoding of textual information, although links to graphical images and other web components can exist in the document; methods, and technologies to enable this are discussed further in Sections 3.4, 3.5, and 3.7. The final interpretation of the document occurs in how the application interprets these tags, when parsing the file, before display.

The original version of HTML allowed for basic stylistic adaptation of text, for example levels of heading, and links to other documents. By following a specified structure (shown in Figure 3.2) the developer can control how and where the application displays the information contained in the document.

```
<HTML>

<HEAD>
  <TITLE>...</TITLE>
</HEAD>

<BODY>
.
.
.
</BODY>

</HTML>
```

Figure 3.2: Basic structure of an HTML file

Figure 3.2. An HTML file is an ASCII text file that is parsed by a browser before being displayed to the user. Various tags define how the text and images are displayed on the screen.

Firstly, the `<HTML>` tag confirms to the application that it is an HTML document being loaded. Various extensions to this indicate which version of the specification was used to encode the document. Following this, the next piece of information, `<HEAD>`, contains header information for the document, for example; `<TITLE>` which most applications display in the title bar. For all the tags, as seen in Figure 3.2, there is a corresponding closing tag that marks the end of that piece of information. These are represented by the `'/'` at the start of the tag, for example; `</TITLE>` closes the `<TITLE>` tag. The `<BODY>` tag encloses the main Section of the page to be displayed by the browser. An HTML file has an inherent hierarchical structure produced from the tags. For syntactically correct HTML, it is important that encapsulated tags are opened and closed in the correct order.

Initially, HTML documents were created using simple text editors. Today, dedicated editors, and conversion tools from many packages, for example, word processors, spreadsheets and databases, exist that allow WYSIWYG* editing for web documents. However, to take advantage of many of the newest tags and technologies related to HTML, it is necessary to code pages by hand. This is due to the speed at which the browsers are including new tags, and the increased usage of beta browsers. With the new techniques to advance HTML, as detailed in the following sections, this research suggests this move to developing tags will slow.

As pure HTML only allows for the storage of textual information, its use for spatial information as required by the VHN project is limited. Although graphics can be included in an HTML document, as explained in the following sections, this is only as a link to another document. Where HTML is powerful is in the ability to use it as a document format for displaying information from distributed sources. Many database servers can now respond to Internet based queries and return results in the form of HTML documents. The geographic information can therefore benefit from HTML as a format for displaying data, rather than as a format for storing data.

The VHN project uses HTML to control the layout of all the elements of the interface, for example the positioning of the virtual world and the two-dimensional map. HTML is also used to describe the style in which to display the textual information that is returned from the database queries. To allow for interaction with this static information it is necessary to look at the area of the Dynamic HTML formats and the capabilities

* WYSIWYG (What You See Is What You Get) applies to the concept that the document appears in the editor in the same form it appears when viewed or printed. In the particular case of HTML editors few can actually offer true WYSIWYG due to the differences in interpretation between browsers.

they contain for naming sections of the page and controlling these via scripting languages.

3.2 *Dynamic HTML*

The initial development of HTML was for the static display of textual information across a remote network. As the technology has progressed, there has come about a requirement to allow user responses to alter a page. This required the user response to be returned to the server, where the required information was formatted into a new page and returned to the client. The technologies discussed in this section allow the view of the information contained in the page to change and react to user actions without always needing to refer back to the host server. The combined usage of these developments is often referred to under the title Dynamic HTML (DHTML). Currently there is no standard for DHTML, and it is implemented differently in the Microsoft and Netscape browsers. For the purpose of this research and the VHN project, the technology supplied by Microsoft in the Internet Explorer 4 (and later) application is used.

VHN uses DHTML to allow control over the content and positioning of the information displayed on the web page, as well as enabling it to be updated in response to the user actions, and the actions of the components. With HTML alone little control was given to the developer over the final display of the page, as this was left to the application. DHTML allows more complex pages to be developed but can also be seen to restrict the portability of these pages. The technologies of DHTML often require the user to use a specific version of an application due to differences in implementation between companies and versions. For special needs users, for example the blind, the use of DHTML can make the site inaccessible to them as the page moves away from being an organised piece of text to a combination of fragments. One development that could

solve this problem is the Extensible Markup Language (XML) discussed later in Section 3.5. This allows for the information on the page to be tagged more effectively while allowing different views on the same data source using enhanced style sheet technology.

3.2.1 Style sheets

Although HTML allows for the definition of the style with which to display a particular piece of text, the aim is that the style of the document is made separate to the content. This allows the style to become application dependent rather than information dependent. Cascading Style Sheets (CSS) allow for more customisable views, for example, accounting for those with vision impairments. Due to how HTML was developed by software companies (Section 3.1), tags were placed into HTML to allow for stylistic design, for example the tag. The idea of CSS is to allow the document creator to suggest how the document should be displayed as well as give an easier way to give a similar look to a series of documents.

```
BODY {background-color:darkgray;
      color:black; }

H1 {   color:red;
      text-align:center;
      font-size:24pt;
      font-family:Forte,Curlz MT,Comic Sans MS; }

DIV {  position:absolute;
      padding:10,10;
      text-align:justify; }
```

Figure 3.3: Example of a Cascading Style Sheet

Figure 3.3. Cascading Style Sheets contain information, which effect the style and position with which particular HTML tags are drawn in the browser.

The example CSS document shown in Figure 3.3 shows how the various stylistic attributes of an HTML tag can be controlled. It is also possible to create particular class instances of a tag, for example the DIV.shadow class above. This allows the usage of

inheritance, or cascading, in the styles used. If used correctly, CSS is a powerful tool for altering the style and formatting of an HTML document. Figure 3.4 shows the results when this style sheet is applied to an HTML document.

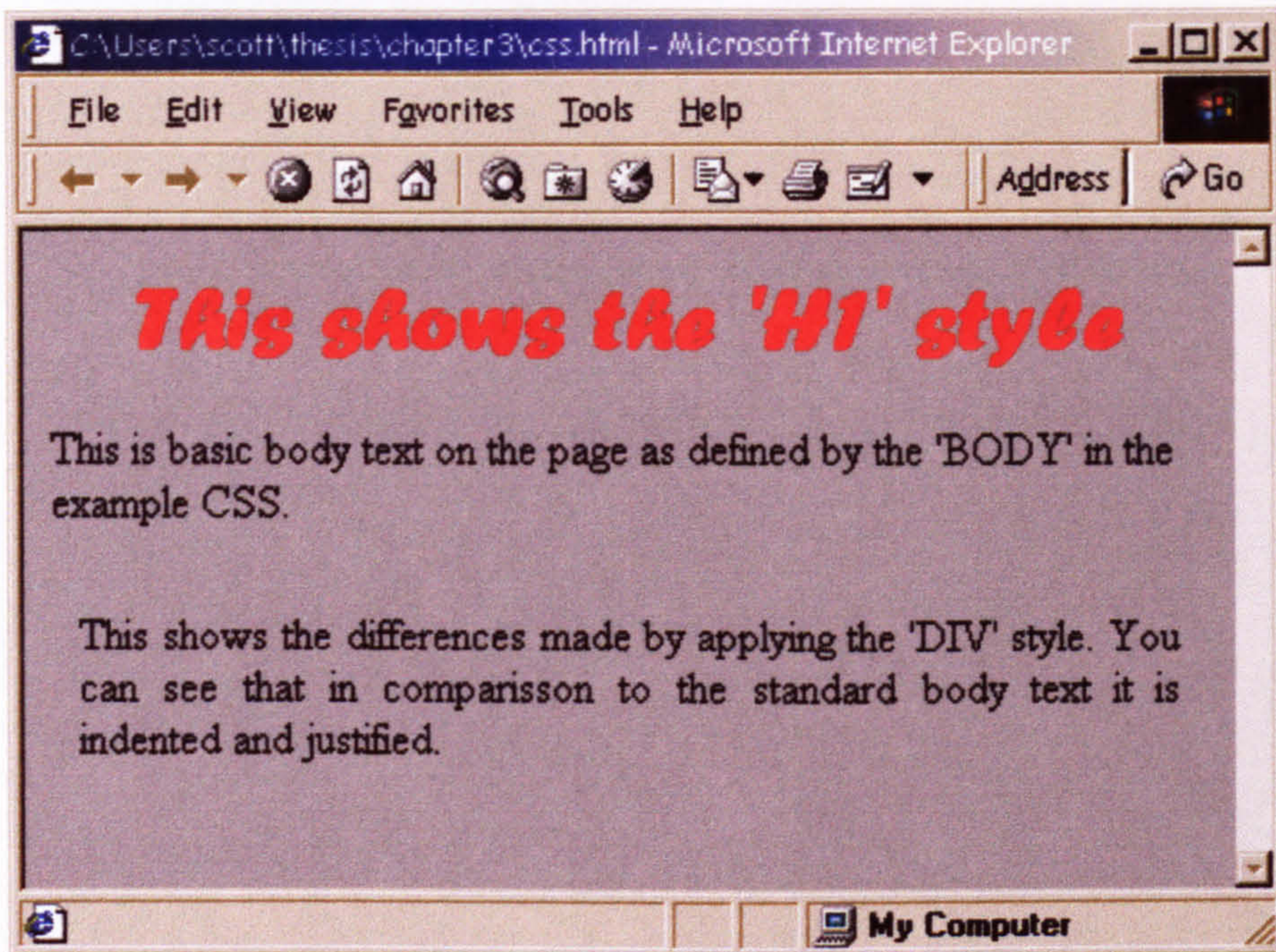


Figure 3.4: Resulting output of a Cascading Style Sheet

Figure 3.4. This shows an example of the resulting output of the CSS as illustrated in Figure 3.3.

Removing the display style from the data, means that the information can more readily be used in a variety of applications. With standard HTML, to display the same information in two different formats requires the duplication of the data. Through the addition of style sheets, the same information can be presented in as many formats as style sheets are written to create. However, style sheets as implemented by CSS are still designed to act on blocks of information, for example whole paragraphs, rather than the individual pieces, such as a sentence or word. It is shown later in Section 3.5.2 how the new XSL standard allows for more control over the individual pieces of information. This is obtained through a stricter separation of the information from the display.

3.2.2 Document Object Model

The Document Object Model (DOM) defines how the document is exposed to scripting languages, for example ECMAScript, by a browser. The idea of the DOM is not restricted to HTML documents and many other applications for example Word and Excel from Microsoft, supply DOM's for their documents. The DOM gives access to the objects contained in the document.

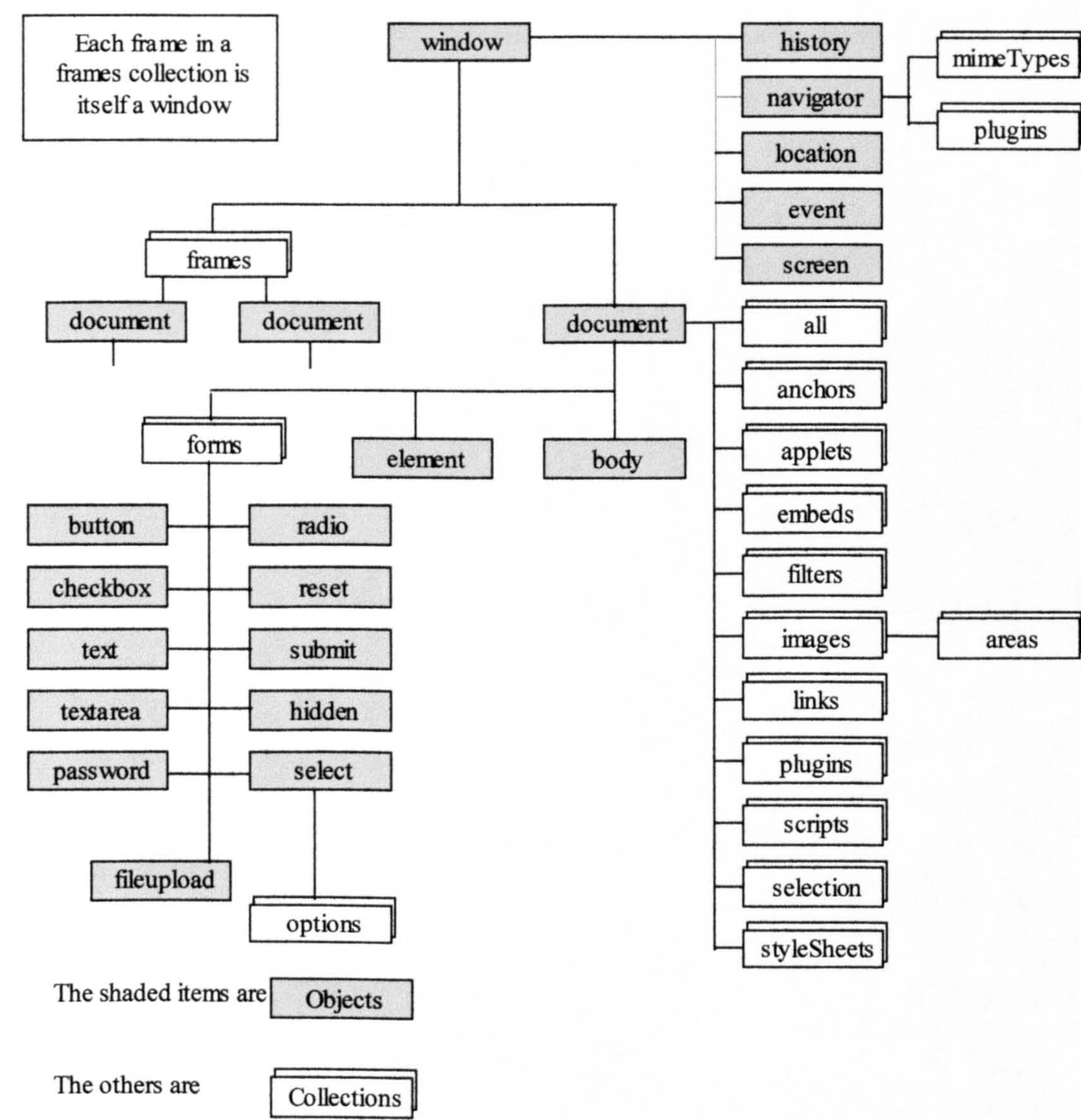


Figure 3.5: The Internet Explorer 4.0 Document Object Model

Figure 3.5. The Document Object Model exposes various parts of the Internet Explorer application and the displayed HTML page to scripting (Homer and Ullman 1997).

Figure 3.5 illustrates the DOM for Internet Explorer 4.0, which includes access for frames, and forms, tags, and any plugin objects referenced in the document. Access is also given to various pieces of information about the application and system that is being used to read the document.

On its own, the DOM does not add to the display and use of the document. Where it is important is in allowing an interface into these areas for scripts. The difficulties come from the current situation where no standard DOM exists for a document. For the case of HTML, the DOM can vary in minor ways between browser implementations. As with most new technologies, until the DOM is standardised it is necessary for developers to adjust their code to the browser viewing the page. For the DOM to be useful to customers of applications, standard top-level DOM's are needed for the various styles of documents, for example, word processors, spreadsheets, and maps. In this way, scripts could be easily ported between applications of the same type. With increased use of distributed documents it is important that the DOM does not allow access to parts of the system that could cause harm. The DOM supplies VHN with a way of accessing the various sections and components on the web page, and the interaction of these through scripting.

3.2.3 Scripting

As explained above, through exposure of the DOM it is possible to use scripting to control the appearance of the object. For security reasons scripts in Internet browsers are only allowed access to the current document, and cannot create or read files on the host computer. However, a lot of information is still available from the browser about the user, for example email addresses, and previous documents visited. For this reason, many users turn off scripting except for pages received from trusted sites.

Scripting allows the document to interact with the user in response to user initiated events, for example mouse movement and clicks. This speeds up the user interaction as more action can occur at the client end without having to return data to the server. Scripting can also be used to bind together the various objects included in the page. For example, the text displayed in a control can be altered depending on the selection from a list. Linking together the responses and the data of separate objects in this way can create powerful applications.

Various scripts exist in the VHN application to allow for the interaction between the various components that are displayed on the page. These are designed to respond to events, rather than follow a straight flow of functionality, so that they react to the user actions. By using the DOM information displayed on the page can be updated and altered as the user investigates the various areas.

3.3 Displaying Graphics in HTML files

The previous methods have been more concerned with the display of textual information on an HTML page. To allow the inclusion of geographical data it is also necessary to be able to display images and models within the HTML document. The simplest way to display graphics in an HTML document is by the use of the tag in the body of the document, as illustrated in Figure 3.6.

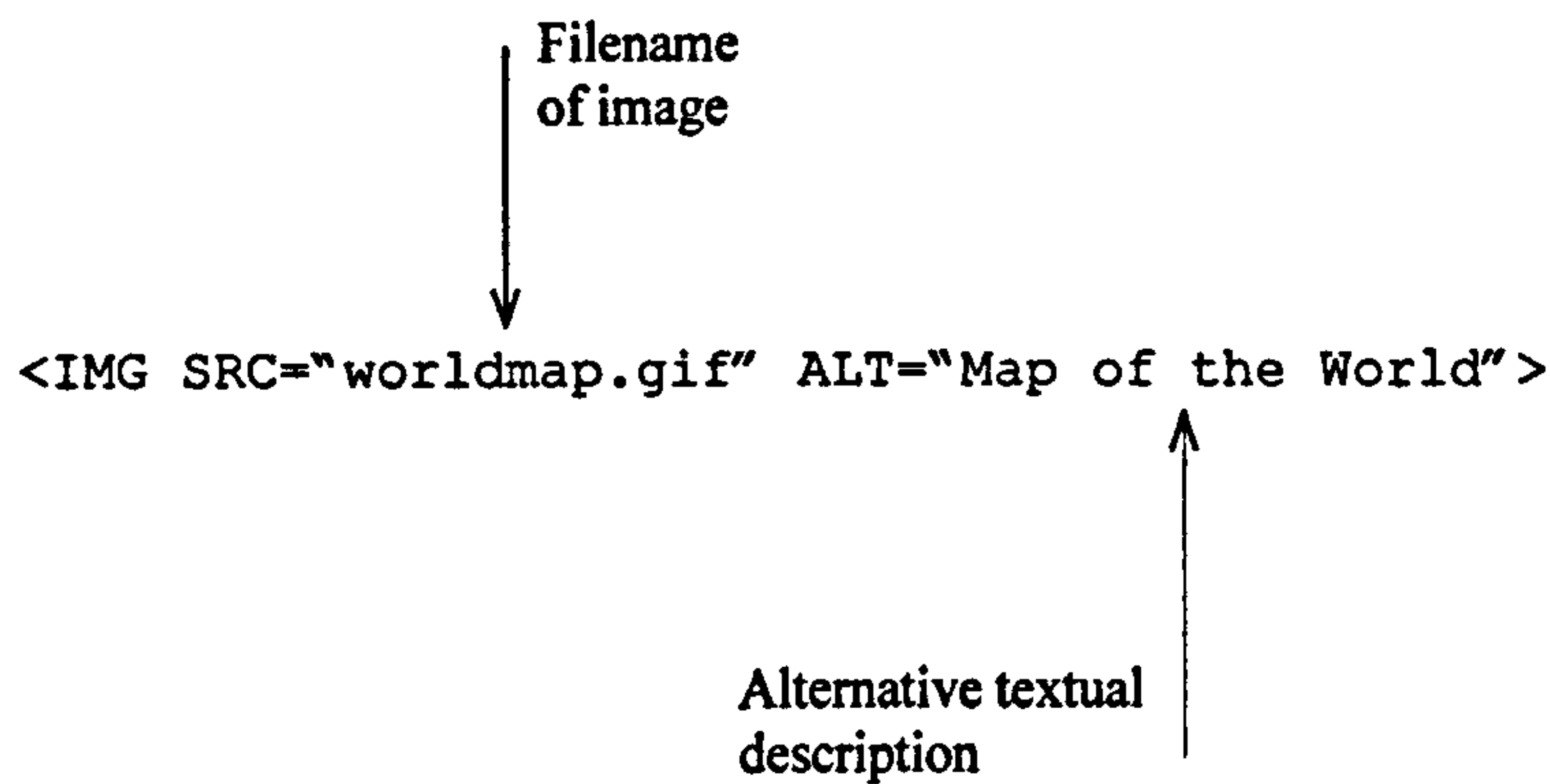


Figure 3.6: Example of `` tag format

Figure 3.6. The simplest way to include graphical information in an HTML document is by use of the `` tag.

The `` tag allows for an image to be displayed in a document, or for an image to act as a hyperlink, if enclosed by the `<A>` anchor tag. The alternative textual description is optional, but is usually given. This textual description is for display while the image is loading, if the image is not found, or if the browser is set up not to load images. In specialised browsers, this tag may be read out when the user places their cursor over the image to aid partially sighted or blind users.

To allow more interactions between the user and the image, HTML image maps can be used. Here, when the user clicks on the image, the location is processed to decide the action of the hyperlink. For example, the world map in the HTML above may be an interface to information on the different countries or continents, as demonstrated in Figure 3.7. Image maps can be processed by the server or by the more advanced client browsers. The trend towards using client processing allows for further information, found in the `<ALT>` tag, to be reported back to the user as they move the cursor, for example, the final location of the hyperlink.


```
<IMG SRC="worldmap.gif" ALT="Map of the World"
USEMAP="#world">
.
.
<MAP NAME="world">
  <AREA SHAPE="RECT" COORDS="2,2,87,31"
    HREF="britain.htm" ALT="Information on
    Britain">
  <AREA SHAPE="RECT" COORDS="2,35,87,64"
    HREF="germany.htm" ALT="Information on
    Germany">
</MAP>
```

Figure 3.7: Example client-side image map

Figure 3.7. Modern browsers allow for the creation of client-side image maps. These allow for areas of the image to link to further documents.

The usage of image maps still only allows for the display of raster information in the browser. For vector based large-scale data, the vector map information must be externally converted to a raster format before it can be accessed for display in an HTML document. This means that the data loses quality, resolution, and information when it is displayed to the user. Where all that is required is a final visual output, or restrictions exist on the access to the raw data, this solution may be satisfactory. For actual GIS functionality then it is preferable to be able to receive the data in its native format.

The VHN project makes use of the image map functionality to integrate the two-dimensional map with the three-dimensional world. Various hot spots, or map areas, are set up on the image, allowing the user to select these to move to locations in the three-dimensional world or display photographs taken from that location. The inherent difficulty of using image maps comes from the fact that the hot spots and the geographical data are two separate pieces of information and therefore both must be updated if the graphical data alters. For GIS applications, it is better for the graphical data itself to be able to respond to the user events. For this, an object based graphical

representation is required and this can be supplied by the VRML for three-dimensional data, or the developing vector markup languages for two-dimensional data.

3.4 Virtual Reality Modelling Language

The initial drive behind VRML was not just to create a format for viewing three-dimensional worlds on the Internet but for VRML to replace HTML as the user interface. This would allow the creation of the 'cyberspace' referred to in many works of science fiction. Users will be presented with virtual rooms of information, rather than the current view of individual web pages. The advantages of this three-dimensional user interface are that it is more in line with our standard view of the world, and that how these rooms are designed is entirely up to the imagination of the owner. However, this therefore has the capability of turning what may currently be a well-structured document into a poorly designed virtual world. Turner (1998, page 36) believes

“In the near future, 3-D visualisation will be enormously affected by VRML’s explosive growth.”

Currently, this aim has not come to fruition, with very few sites employing VRML for their interface. This may be due to the hardware requirements required for usable desktop VR. As the basic specification of home and business PCs increase, it is liable that the usage of VR will too.

VRML came about in the spring of 1994, following the World Wide Web Conference in Geneva. The initial draft proposal was created by November of that same year, and the latest specification is VRML 97. In that time, VRML has moved from allowing a static description of a scene to a simple virtual environment language. There is, however, still

a long way to go towards full VR across the Internet. As with HTML it is necessary to have a browser to view VRML files. This can be either a stand-alone browser such as CyberPassage from Sony, or an HTML browser plugin, as in Viscap from Superscape. As with HTML, various 'flavours' of VRML exist, containing browser specific enhancements to the language to allow for further functionality.

The files created for handling spatial data in VRML are termed 'virtual worlds'. The construction of such files representing an environment or other spatial data can be undertaken using a number of packages which can act as authoring tools for interactive scene creation or enhancement of existing spatial data. However (as with HTML), the most 'advanced' VRML has to be hand-coded in a text editor. When dealing with imported shapes from standard CAD and drawing packages (rather than shape primitives initiated within VRML 'writers') neat, compact VRML is difficult to achieve. The current implementation of VRML is aimed towards the use of geometric primitives to represent worlds. Little functionality is included for the compact use of co-ordinate based data, relying on objects to be created from lists of positions.

In a similar way to HTML, VRML allows for 'hyper-links' within an image. Using such a tool, it is possible to construct a 'walk' through of a map, and link into further (connected) worlds, images, text and animations. Unlike HTML, however, the 'virtual reality' nature of VRML allows for user control over the choice and position of the viewing location. Solid, three-dimensional, objects can be looked at from any position and in any perspective. Two-dimensional images can be viewed from any distance. Thus, map scale can be interactively changed to suit the viewers' requirements.

Associated with this is the ability a creator has within VRML to vary the level of detail (LOD) of objects within the view, dependent on the distance from the viewpoint. In

effect, cartographic generalisation can be incorporated into the map image. This generalisation is not automated; the levels must exist as separate objects, within the content of each world. In addition, the points at which levels of detail are automatically changed and new images replace the current ones need to be explicitly stored. In effect, the VRML tools allow for a direct, although easier, replication of the generalisation sequence of paper mapping, with differing maps being called up and viewed as the requirement to change scale is logged. Generalisation may be assisted with the introduction of tools such as LODestar from the Technical University of Vienna, discussed in Section 5.5.

Fairbairn and Parsley (1997) suggest that such generalisation is a function of the zoom capability of any cartographic visualisation tool. Further cartographic issues arise from such a capability; image zooming does not have a direct equivalent in paper map use and has to be applied with care (von Wyss 1996). In addition, animation also poses interesting new demands on the cartographic designer; there is as much need for artistry, intuition, and imagination in this area as in static mapping. Consider the example virtual worlds suggested by Collinson (1997) and illustrated in Section 2.6.

Fairbairn and Parsley (1997) continue to state that during human-computer interaction, animation can divert attention towards the most rapidly changing parts of the image, to the detriment of the remainder. Animation, zooming and panning are all inherent in the use of VR, although the user may accept these almost sub-consciously in their operations. The combination of these techniques for constantly changing the position of a virtual world may lead to information overload.

Movement around the image is governed by mouse operations, and can be considered as a weakness in efficient interaction with a scene. Navigation within a three-dimensional

environment using a two-dimensional tool is difficult, and quite often the intended movement of viewing position is not easy to achieve. Some viewers allow for a switch, using a menu system, between classes of movement such as back/forward and spin vertical/spin horizontal. It is also possible to enable the object(s) to move, rather than to change the viewing position, and this may seem intuitively preferable for some users. The technology of immersive and transparent VR has matured so that an interaction device such as a three-dimensional mouse or a space ball, common input devices in the CAD industry, can potentially be used with a desktop VR system. The transition from two-dimensional geographic data into movement in VRML is also hindered by the lack of a geographic co-ordinate system in the language. The default implementation is for worlds to be centred upon the origin, (0, 0, 0). This means that both the data sets and the user's position have to be transformed before display.

Fairbairn and Parsley (1997) conclude that an assessment of VRML for cartographic purposes can address many issues; world creation and reference systems, cartographic generalisation, navigation and the user interface, display and animation, and applications areas. The creation of worlds for VR parallels, in many respects, the work of the cartographer in representing reality, with VR able to utilise spatial data collected for map-making, or stored in map or digital spatial database form. This suggests that it is perhaps with the map user that the most enthusiastic applications of VR will find fruition.

From this research, it appears that VRML is finding more favour as a simple transfer format than a storage format. Because of the unpredictable nature of developments in computing, the role of VRML for spatial information is uncertain. Future developments to the Internet based languages, discussed in the following section, suggest that the use

of Internet technologies in desktop applications will increase. In the development of VHN, an alternative file format, Superscape's Viscap was used. This was necessary due to the size and complexity of the virtual worlds that were created. It was felt that with current limitations on bandwidth produced by the use of 56K modems that the file size should be kept as small as possible.

3.5 Future Directions

Presently the language used to transfer data on the web is undergoing a change towards the Extensible Markup Language (XML) which, like HTML, is a subset of SGML. XML was originally designed as a way to simplify the power of SGML for the web. Unlike HTML, where the tags mainly define how the data appears on a page, XML is about making data into information. In many ways XML is truer to the original concept of HTML and answers many of the complaints of HTML, for example removing all stylistic and positional information from the document.

XML is not described as the next version of the HTML format; it is a change in concept of how data is shared on the web and in general. To understand how and where XML fits into electronic documentation it is necessary to consider the elements that make up a document. In this way a document can be separated into three divisions

- content – the data components, for example; words, pictures, records, etc
- structure – the organisation and interrelations of the data components
- presentation – the style and processes that are applied to the data

Currently, one of the downfalls of HTML is that it attempts to handle all of these elements of a document. Recent improvements such as CSS, discussed earlier in Section 3.3.1, have aided by removing some of the presentation aspects. However,

many HTML creators (both human and automated) still prefer to misuse tags, for example using `` for indenting, rather than the intended use; for indicating an unordered list. However, XML purely defines a means for storing structured content in order to add intelligence to data, not style. This facilitates database exchanges, client-side processing, access to richer data, and better data management. The related Document Type Declaration (DTD), Extensible Style Language (XSL), XLink, and XPointer specifications (detailed later) concern the presentation and processes of the data. As mentioned in Section 3.2 this increased separation of the data from the display, and the inclusion of computer readable intelligence as to the information it contains, increases the usability of the data as well as the accessibility. This allows computerised techniques to be used to extract the required information from the data file, rather than relying on the final interpretation by the user.

To look at, XML is very similar to HTML, mainly as they are both subsets of SGML. However, the tags used to markup the data in the document can now be decided by the author. In HTML, these tags mainly define the style of the HTML document appearing on the page, for example `bold` and `<I>italic</I>`. This is fine for display as a web page (Figure 3.8) but means that the data loses the information that it conveys to the user.


```

<PRE>
<B>Sold to:</B> Scott Parsley
<B>Sold on:</B> 1999-03-02
<B>Item 1:</B> Thompson, Nigel, <I>"3D Graphics Programming for Windows 95"</I>, £32.99
<B>Item 2:</B> Bannister, A, <I>"Surveying Sixth Edition"</I>, £16.99, ISBN 0-582-07688-9
<B>Item 3:</B> Parker, David, <I>"Innovations in GIS 3"</I>, £15.99
<B>Item 4:</B> Rice Krispies Square, Chocolate, £0.45
</PRE>

```

```

Sold to: Scott Parsley
Sold on: 1999-03-02
Item 1: Thompson, Nigel, "3D Graphics Programming for Windows 95", £32.99
Item 2: Bannister, A, "Surveying Sixth Edition", £16.99, ISBN 0-582-07688-9
Item 3: Parker, David, "Innovations in GIS 3", £15.99
Item 4: Rice Krispies Square, Chocolate, £0.45

```

Figure 3.8: Example HTML and related display

Figure 3.8. The above example HTML shows how a customer order may be marked up for display in HTML.

Where XML supersedes HTML is in the ability to describe the information, allowing the data to be interpreted by machine as well as the reader. This has the trade-off that the file itself becomes larger, and therefore a sensible approach to markup has to be taken. Figure 3.9 illustrates the same data transferred as before with the HTML representation, but here more information is included about the data. For GIS, this allows the XML document to be searched in a similar way to a database file in order to locate data based on matching criteria.


```

<order id="scottp">
  <sold-to>
    <person>
      <firstname>Scott</firstname><surname>Parsley</surname>
    </person>
  </sold-to>
  <sold-on>1999-03-02</sold-on>
  <item>
    <price>32.99</price>
    <book>
      <title>3D Graphics Programming for Windows 95</title>
      <author>Thompson, Nigel</author>
    </book>
  </item>
  <item>
    <price>16.99</price>
    <book>
      <title>Surveying Sixth Edition</title>
      <author>Bannister, A</author>
      <isbn>0-582-07688-9</isbn>
    </book>
  </item>
  <item>
    <price>15.99</price>
    <book>
      <title>Innovations in GIS 3</title>
      <author>Parker, David</author>
    </book>
  </item>
  <item>
    <price>0.45</price>
    <confectionery>
      <brand>Rice Krispies Square</brand>
      <flavour>Chocolate</flavour>
    </confectionery>
  </item>
</order>

```

Figure 3.9: Example of a XML file

Figure 3.9. The XML version of the same data as in Figure 4.6 adds more information by using the tags to convey this information, not how to display the data.

To understand and process the extra information in an XML file, further related documents are required. These documents are used to explain the content of the tags, as well as decide the visual display of the information. The following sections describe the content and format of these files.

3.5.1 Element Definition

In order to be valid, rather than just well formed, a XML document requires a DTD. Current HTML files should contain a reference to an SGML DTD; in the future, this will become an XML DTD. The DTD defines the tags that can be used in a document and what these tags can contain, as depicted in Figure 3.10. Whereas, currently the HTML DTD is decided by the W3C the XML DTD is decided by the user, or the community within which the document has relevance.

```
<!ELEMENT order (sold-to, sold-on, item+)>
<!ELEMENT sold-to (person)>
<!ELEMENT person (firstname, lastname)>
<!ELEMENT item (price, book | record | confectionery)>
<!ELEMENT sold-on (#PCDATA)*>
<!ELEMENT firstname (#PCDATA)*>
...
```

Figure 3.10: Part of the DTD for the earlier XML document

Figure 3.10. The DTD is used to describe the types of data that can be included in a valid XML document using that DTD.

Figure 3.10 shows a DTD that states that, an order element must contain a sold-to and sold-on element, but can also then contain as many item elements as required. The order of elements is enforced in an XML document, though it can be altered when the information is displayed. At the lowest level, an element contains PCDATA, i.e. parseable characters. Although this may look long, it is necessary to remember that a DTD only has to be defined once, and is then referenced in all documents. The aim is that user communities will create DTD's for the interchange of their data. For example, proposals for mathematical, chemical, musical and graphical DTD's have already been submitted to the W3C for review.

Currently there is no work being done on the creation of a general DTD for the transfer of spatial data in XML. It is feasible that one of the specifications mentioned later, in

Section 4.2, may be used to create the DTD, as well as building on any related DOM's. Through the flexibility of XML as a markup language, it will not only be possible to transfer textual data by these means but also for the transferral of graphical data. The limit on the information that an XML document can contain is in the ability of the application used to parse the document.

3.5.2 Styles

The actual display of the XML document may be the same as in Figure 3.7; the difference is that the style is no longer stored in the document. This allows separate styles to be applied to the data depending on its usage, for example viewing or printing. Styles are handled by the usage of the XSL, embedded in the DTD (or an applet), by reference to pre-existing namespaces or by reference to the DOM. The use of XSL gives similar capabilities to that supplied by CSS, but also adds more. For example, Figure 3.11 shows part of the XSL document for the earlier XML detailing how to display the person element on the screen.

```
<xsl:template pattern="person">
  <xsl:sequence>
    <B><xsl:process select="lastname"/></B>
    <xsl:text>, </xsl:text>
    <xsl:process select="firstname"/>
  </xsl:sequence>
</xsl:template>
```

Figure 3.11: Part of the XSL for the above XML document

Figure 3.11. The XSL specification allows for a similar functionality to that gained from CSS except that now the order and arrangement of elements can be controlled as well as the style.

XSL not only allows for the specification of the style with which the element is displayed, but also the order and any other text or elements to place around the original

element. Figure 3.12 shows the earlier example of a person element, and the subsequent display using the above XSL statement.

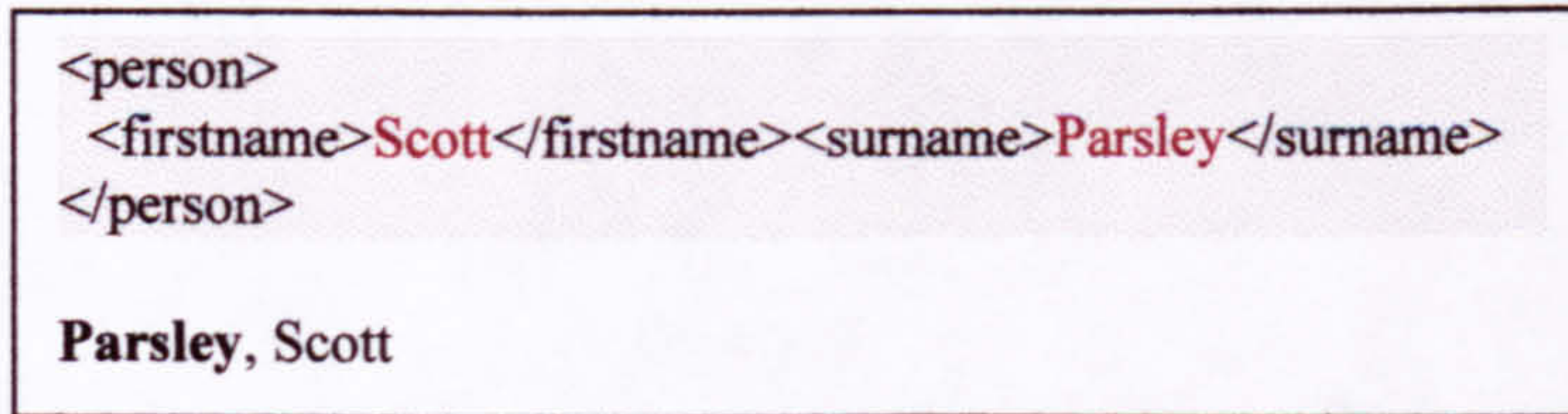


Figure 3.12: Example of applied XSL

Figure 3.12. By applying the XSL from the earlier example to the person object it is possible to see how the XSL for an XML document can control how an element is displayed.

This usage of XSL to control the formatting and visibility of elements means that entirely different views of the same document can be created. As with the DTD, XSL documents are linked to the XML document, meaning that they can be reused company-wide. Therefore, a company may create an XSL document for standard viewing, vision impaired viewing, public viewing, printing, or any reason they choose, yet the actual data is only stored once. The use of XSL also means that users can specify in their browsers their own XSL document with which they wish to view any XML. This can be seen to be similar to the existing case of separate database views in GIS for differing users. In the VHN project, this allows public and private views onto the same data set via the XSL describing the content and format of the data to display. This reduces the requirement on the project to hold the data twice, as content to display to the public and as content for research.

3.5.3 Linking

Linking in XML is an enhancement on the functionality currently available with HTML. HTML has three tags to link to another document; the `<A>` (anchor) tag, the `` (image) tag, or the `<OBJECT>` tag. These links always cause the whole

document to be loaded into the browser, either to replace the existing documents (anchors), or to displace the data in the existing document (objects and images). An anchor reference can contain a reference to an anchor tag in the new document to scroll that section of the document into view on loading; this can give the effect of referencing a specific part of a document. This, however, requires ownership of both documents to include links to new sections of the document.

For XML, this has been taken further by the implementation of the XLink and XPointer specifications. These allow for any tag in a well-formed XML document to be referenced and only that information to be returned from the document. Therefore, it would be possible to create an XML document that was entirely made up of clippings of other documents.

As any well-formed XML document can be represented by a tree structure, it is possible to navigate around this tree from any hook into the tree. The below example, Figure 3.13, shows how a XPointer can be formed to reference one of the objects from the earlier XML document. In the example the hook, ID="scottp", is used to get hold of the main order element. Then the tree is navigated to get to the second item element child, and from there, to the book element. What is returned is just the requested element, speeding up data transfer and removing unnecessary data the user did not wish to see. If no hooks exist in the tree then the XPointer reference can start at the top of the tree and traverse the whole tree. Hooks allow for a more user-friendly approach, and shorten the length of the pointer.


```
http://shop.com/orders.xml#id("scottp").child(2,item).child(1,book)

<book>
  <title>Surveying Sixth Edition</title>
  <author>Bannister, A</author>
  <isbn>0-582-07688-9</isbn>
</book>
```

Figure 3.13: Example of linking in XML

Figure 3.13. The ability to link to a part of a document and only retrieve that Section of the document is one of the major advantages of linking in XML over HTML.

As with HTML, XML is likely to find more favour with the spatial community as a data transfer format rather than a storage format. The ability to include information about the data being transferred means that any knowledge or relationships the data contains can be recovered. The ability to display sections from separate XML documents in the same view, by use of the XPointer, format allows for the display of results from multiple database queries on remote servers to be displayed coherently in one document.

3.6 Vector Mark-up Languages

Of particular interest to this research and the VHN project is the ability to represent two-dimensional vector data in XML, which will allow the display and editing of vector data across the Internet. The main specification proposed by the W3C is Scalable Vector Graphics (SVG), developed from earlier memos to the W3C suggesting the possibility for vector mark-up languages. There is also related work in the three-dimensional area with a proposal to convert VRML into XML, called 3DXML, by the web3d consortium.

The SVG specification was proposed by a range of companies concerned with graphics, for example; Adobe, Autodesk, Xerox, Visio, Macromedia, Corel, and Microsoft. In

November 1998, a proposal of SVG requirements was created by a working party containing representatives from the above companies. The most recent draft of the specification was made available for public comment on the 11th February 1999. Now the specification is in a preliminary form, though many of the sections contain proposed syntax.

Unlike a raster representation of a vector graphic, SVG will allow the user full access to the definition of the graphic through the DOM and a scripting language. In this way, events can occur when the cursor is moved over segments of the graphic, allowing sections of the graphic to change shape or style depending on a user response.

An example, from the earlier Vector Mark-up Language (VML) specification on which SVG is based is shown in Figure 3.14. When processed this will produce a star shape. The `<V:SHAPE>` tag informs the browser that the following data is a VML graphic shape. Further information contained by the tag defines the type of line, the colours, and the points of the shape. The definition of objects in the SVG language is similar.

```
<V:SHAPE STYLE='top:0; left:0; width:250;
height:250'
STROKE="true" STROKECOLOR="red"
STROKEWEIGHT="2"
FILL="true" FILLCOLOR="green"
COORDORIGIN="0 0" COORDSIZE="175 175">

<V:PATH V="m 8,65
1 72,65,92,11,112,65,174,65,122,100,142,155,92,
121,42,155,60,100
x e" />

</V:SHAPE>
```

Figure 3.13: Example VML of a star shape

Figure 3.14. VML allows for the creation of vector graphics in a web page. Although currently designed more towards desktop publishing graphics, there is no reason why geographical data could not be coded in this way.

Initially aimed at desktop publishing style graphics, there is no restriction to encoding geographic data. Converters would be necessary to change current data files, for example the National Transfer Format (NTF), into VML. VML is still confined to two-dimensional shapes, although these can be arranged on layers in XML, giving a similar appearance to layers, as in current GIS systems. In order to display three-dimensional data it will be necessary to use the 3DXML which is being developed.

The display of vector graphics in an HTML, or XML, document removes the restrictions of raster images mentioned earlier. Here, spatial data can be transferred in a vector format allowing information to be attached to the individual graphics objects in a similar way to current GIS applications. As the technologies mentioned here are human-readable, it follows, as before, that SVG and its related formats will be used as transfer formats.

3.7 Summary

More applications are now being designed that allow their data to be distributed across the web. With the new standard of XML and the related standards, now being released, the potential for the web is unknown. Many large commercial companies are putting more backing behind web based technologies. For example, Oracle will allow Oracle 8i to serve data in XML and the new version of MS Office (Office 2000) uses XML as an underlying common data format for all the applications. In many ways the Internet is achieving what standards bodies have been trying to introduce for the past few years, a common data transfer format. Already many packages can produce HTML with the increased flexibility of XML, and with the inherent ability to store any data, this process can only continue. This will lead to better tools for processing the data into information, as the previous restrictions of being locked to one supplier to access data

will be removed. Given how much the web has grown in the past eight years, then it is impossible to say what languages and applications the web will be using in the next eight.

With the web tools available, supplied by DHTML, complex applications can be made from simple components created to perform specific tasks, allowing the developer to include only the functionality their web page requires rather than a complete GIS. Currently the ability of GIS across the Internet through web interfaces is restricted by the lack of these components. The developing web is being influenced by the needs of multimedia and Internet commerce, the GIS community also needs to get involved to ensure the technology develops for spatial information too. Chapter four examines the current developments in the Internet and component systems that may supply the underlying infrastructure that is required to allow the implementation of a GIS that has a fully web-based interface.

The way forward for GIS is not to continue with the monolithic systems that we have today where the entire system is a single application. By employing the ideas of object-orientation in both applications and data stores, GIS can take advantage of the new ideas in areas such as visualisation, while remaining with the legacy of two-dimensional systems. Much of the delay may be attributed to the fact that to add new functionality to a system requires the upgrading of the system rather than an extension. A move to a more open system will allow others to build on what is already supplied. The ideology of the VHN project is to be able to create a simple interface application into which components can be embedded to create the functionality. For this to be possible, the idea of plug-and-play has to be extended to software as well as hardware. This concept is dealt with more in the following chapter.

4. Component Systems

At the current time, a major trend in computing is towards component systems. Often referred to by the terms open, object-oriented and distributed, these systems allow for applications and data to become separate modules, or objects, that can exist anywhere across a network of computers. A component system requires an interface to the objects, the creation of a network over which objects can move, standards for how these objects are named, and standards for how these objects will communicate.

The standards mentioned here do not have to be made by formal standards institutions, such as the International Standards Organisation (ISO). Many standards in the computing industries today are referred to as *de facto* standards, for example, the digital exchange format (DXF) from Autodesk. With the speed at which technology and research in the computing industry is growing now, it is becoming more common for software manufacturers to mould the future, rather than the standards bodies, as has been seen with HTML (Section 3.1). Even when standards are introduced, it is not always possible to get them accepted by the computing community. For example, the software to read and create National Transfer Format (NTF), BS7567, files is still an addition to, rather than an integral part of, a GIS. Currently, more software allows for the exchange of spatial information by either DXF or the ESRI formats than NTF.

Chapter 3 detailed how the web can be used to create an interface to components and the data and applications that they contain using the example of the VHN interface. This chapter follows on to explain the underlying technology required to make the web work to its fullest. Beginning with an overview of the network technology, in particular the development of the Internet, the chapter continues to look at emerging object-

oriented standards that will allow for distributed applications and data. These underlying technologies of the web allow for the creation of a multimedia interface to intelligent remote data stores exemplified by the VHN project.

4.1 Networking Technology

The idea of the Internet was to create a decentralised computer network, whereby every computer was connected to each other. In this case, if one of the systems suffers a failure, the others would still function. Originally, these outages were expected to come from bomb attacks in the days of the cold war: nowadays there is more threat from accidental damage to the actual cables.

The first two computer communication nodes, UCLA and the Stanford Research Institute, were set up in the autumn of 1969. By the end of the year, four other nodes were set up on this infant network, the ARPAnet, named after its sponsor, the Advanced Research Projects Agency. The decentralised structure of ARPAnet made expansion easy, with other computer networks from various organisations linking themselves to the ARPAnet by use of cable and radio connections. With the advent of satellite transmission, the first international connections were made with the University of London and the Royal Radar Establishment of Norway in 1973. This paved the way for the global networking between computer systems from different continents that we have today.

In 1979, the Computer Science Research Network (CSnet) was set up by the National Science Foundation (NSF). This network was connected to the ARPAnet through a gateway, and was used for the collaboration of research scientists. CSnet allowed the sending of electronic mail (e-mail), reading of newsgroups and transfer of data files. By

the early 1980's, the NSF had created its own network, the NSFnet, for the exchange of information for educational and intellectual purposes, and this attracted much attention from educational institutions and libraries. The Joint Academic Network (JANET), which now links together the majority of UK academic institutions, was established in 1984. Access to the NSFnet remained limited even after it became the backbone of the Internet when both ARPAnet and CSnet were shut down in 1990.

Until 1990, the only way to connect to the Internet was by a direct connection. A company called The World (<http://world.std.com>) that became the first commercial provider of dial-up access, changed this. Along the way, various methods for accessing information across the Internet were developed. The most useful of these, which are still used today, are FTP, Telnet, and Gopher.

In 1992 access to the Internet changed. The NSF created a separate entity known as the Advanced Network and Services to manage the NSFnet and, with that, the Internet was opened to all interested parties. Over time, with the increased acceptance and availability of the web, the amount of traffic moving across the NSFnet grew, as shown in Figure 4.1. In March 1995, the NSFnet was officially decommissioned. Since then, the number of networks that comprise what is referred to as the Internet has continued to grow exponentially, along with the traffic they carry.

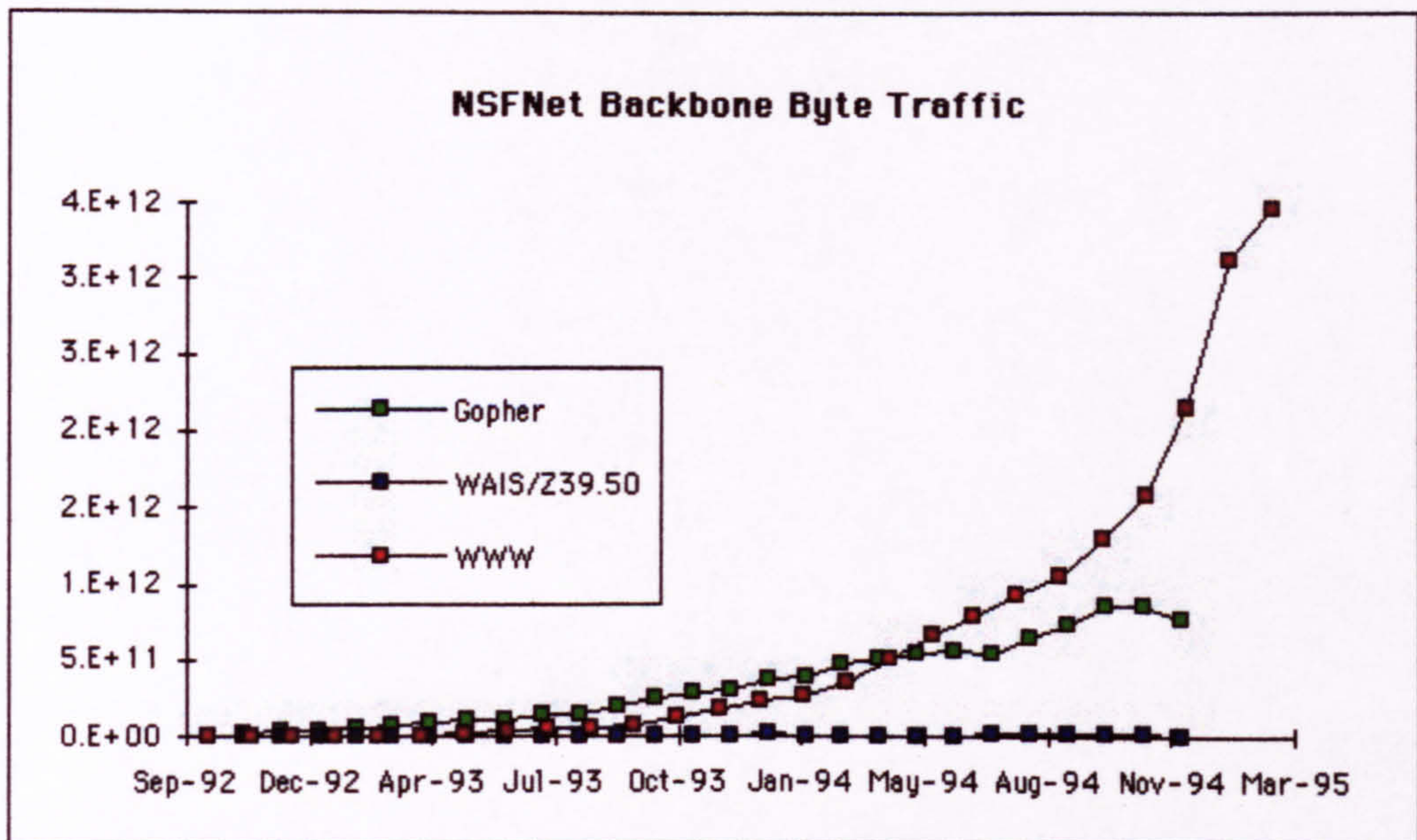


Figure 4.1: NSFNet traffic

Figure 4.1. With the acceptance of the web, traffic across the NSFnet increased exponentially (ZDNet 1998).

This opening up of the Internet came at the time of another breakthrough, the advent of multimedia on the Internet through the web. The advantage of the web, over earlier protocols, is that it uses hypertext and multimedia techniques to make the web easy for anyone to roam, browse, and contribute information to. With pictures, sound and video, it is less daunting than trying to move around in a command shell environment. Technologies that make up the web were discussed further in Chapter 3.

Today, the Internet is still growing in terms of size and number of connections. It is estimated that there are now about 50 million Internet users worldwide, from as many as 100 countries (ZDNet 1998). No one owns or controls the Internet; rather, administration takes place at each of the autonomous networks linked to the Internet. Thus, there is a prevailing sense of anarchy about the Internet. Nevertheless, it is also referred to as a self-governed democracy, where information, good and bad, can be exchanged freely.

The number of methods available today for communicating information across the Internet has also grown. The exchange of data can be either asynchronous or synchronous. AGOCG (1998) lists possible examples of these two categories of communication as

- asynchronous - Email, Newsgroups, Mailing Lists and the Web
- synchronous - Chat, Video Conferencing, Shared Whiteboards and Multi User Dungeons

The availability of synchronous communication has meant that it is now possible for researchers to communicate as easily with other organisations and groups across the world as they can with their colleagues in the same office. Applications such as VHN can harness this technology to allow for collaborative brainstorming between groups within a virtual world. In many commercial applications this virtual world may represent an imagined games arena as with many online games, but the same technology can be used to allow groups of researchers to explore and examine a historic environment.

The main protocol used for transferring information on the Internet is the Transmission Control Protocol / Internet Protocol (TCP/IP). Data is sent as a series of packets tagged with an address of a computer to send them to. Originally, users had to remember the series of numbers that make up an Internet address, for example, '128.240.14.61'. However, nowadays name servers are used to convert these to memorable addresses, for example 'bowditch.ncl.ac.uk'. The data packages are sent through the Internet via gateway computers, which connect different Sections of the network. The route and journey time, for a piece of information sent from Newcastle to a computer in Redlands, California is shown in Table 4.1.

Table 4.1: Route to www.esri.com from bowditch.ncl.ac.uk

Step	Time	Name	IP Address
1	5 ms	bedson-gw.ncl.ac.uk	128.240.240.251
2	6 ms	oakwell.ncl.ac.uk	128.240.1.104
3	5 ms	ncl-man.ncl.ac.uk	128.240.1.6
4	8 ms	leeds-core.ja.net	146.97.254.37
5	13 ms	external-gw.ja.net	146.97.254.57
6	13 ms	tglobe-gw.ja.net	193.63.94.85
7	387 ms	gin-ppt-bb1.Telelobe.net	207.45.206.241
8	129 ms	gin-nyy-ac1.Telelobe.net	207.45.199.233
9	178 ms	sl-gw16-pen-2-0.sprintlink.net	144.228.164.61
10	252 ms	sl-bb11-pen-3-1.sprintlink.net	144.232.5.97
11	205 ms	sl-bb1-pen-8-0-0.sprintlink.net	144.232.5.10
12	181 ms		144.228.180.10
13	283 ms	nyc4-br2.bbnplanet.net	4.0.2.178
14	151 ms	nyc4-nbr2.bbnplanet.net	4.0.5.29
15	239 ms	paloalto-nbr1.bbnplanet.net	4.0.2.201
16	240 ms	paloalto-br1.bbnplanet.net	4.0.2.194
17	240 ms	paloalto-cr15.bbnplanet.net	131.119.0.215
18	247 ms	esri.bbnplanet.net	131.119.28.74
19	253 ms	www.esri.com	198.102.62.22

Table 4.1. The route traced across the Internet from a computer in Newcastle upon Tyne to a computer in Redlands, California. Although the route is usually the same outages and heavy loads on the Internet can cause information to be sent via other gateway computers.

If a fault occurs with any of the computers on this route, then due to the nature of the Internet the data packet is sent a different route. It is quite possible, though rare, that individual packets from the same file can be sent different routes.

The Internet and the web interface that has been created to it have allowed applications such as VHN to become a possibility. The Internet allows data to be stored at the location it is created and then shared between groups in the same office, country, or world. As the data becomes more intelligent about what it contains (see following sections), there is the ability to use the Internet to interrogate and compare a vast amount of data that has previously remained discrete. For this to be possible though, standards must be created for how geographic data is defined and queried.

4.2 Object-Orientation and Components

The ideas of object-orientation originate from computer programming languages and techniques. Now these ideas are being translated, not only into software applications, but also into the data that these use. The object-oriented approach may be applied to a system at various levels. For example, there are object-oriented programming languages, object-oriented modelling methods and object-oriented database management systems. The basic idea of an object-oriented data set is to produce 'intelligent' modules of data that contain methods, as well as attributes. For example, Worboys and Hamilton (1996) illustrate that a region may be specified by

- data - a set of points or curves giving the boundary of a region's extent
- methods - the operations that the region supports

An aim of the implementation of objects into GIS is for the development of open systems. This is the creation of individual applications to do the separate tasks of a GIS. For example, the user interface supplied by the VHN project may use, a database system from ESRI, network applications from Smallworld, and a query system controlled by Oracle software. In this way, users could buy the best of each of the modules they required. This also enables the user to purchase only the features required, thus reducing some of the bulk of the current stand-alone systems. Upgrading would be simple, as it would only be a case of either swapping a module or 'bolting on' any new ones.

For this to work successfully it is necessary to think of software applications and data as individual objects. Objects will then talk to each other by way of an object request broker (ORB). This will act as a messenger, receiving and transmitting information to the relevant objects or another ORB on a remote system (Section 4.3). An ORB based

system gives the user software that can be present in various locations, and which can access data in various locations. The distinction between local and remote will vanish, as all objects are accessed in the same way. The only difference is the time delay that could occur across connections. This blurring of local and remote will not be the only effect that is noticed by the user.

As the data sets become objects, it will become possible to embed more methods into the data sets themselves, eventually creating a form of dynamic data with a level of artificial intelligence. For example, it would be possible to create data with the ability to verify itself, removing this functionality from the applications. In addition, data sets will be able to contain version information. Individual objects will be able to clone themselves when they are altered, thus creating a form of temporal database or history of the object. Changes could then be rolled back to recreate data sets at certain periods, or to allow animations of change.

The advantages of this kind of architecture also appear in the view developers have of the system. Applications will become easier to write, as code reusability is implemented. At the highest level of abstraction the users applications become redesigned interfaces onto the underlying objects already present on the system. It would be feasible for a developer to assume certain standard objects already existed on a system, so they could avoid “reinventing the wheel”, which is currently often done. Examples of these standard objects may be text editors or mathematical interpreters. Some standard objects already exist on the Microsoft Windows platform with many applications using the supplied common controls, for example, for print dialogs and text edit controls. Similarly, usage of object-oriented data techniques allows for the inclusion of any form of data, as long as it conforms to the interface standard. This will

enable new forms of data to be added easily into compliant GIS applications, by changing the data not the application.

The following sections discuss standards being defined for how objects that represent data and applications will communicate with each other. The various standards are discussed with their relation to geographic information and the applications that use it.

4.3 Spatial Operator Standards

In the geographical world, real-world entities are represented by abstractions in the computer data model. In order to allow data sets to become distinct from the applications that create them, it is necessary to standardise geographic information. This includes defining what is meant by a certain type of object, for example a line, and how each object can interact with other objects, for example when a line and a face are overlaid. Now each piece of software refers to geometric objects in their own way. What may be a polyline in one package is a chain in another. As it is not definite from the class name of an object what the data actually is, a distributed object environment is not possible.

Various committees are creating standards for the spatial operators, for example the International Standards Organisation (ISO), Conseil Européen pour la Normalisation (CEN), and the International Hydrographic Organisation (IHO). Related work is being carried out on the Spatial Archive and Interchange Format (SAIF) and the Structured Query Language (SQL), as shown in Table 4.2.

Table 4.2: Data Structures

Formal Spaces	SAIF/SQL	IHO	CEN	ISO
Raster Grid TIN / Delauney Spaghetti Node-arc-area	<i>No particular representation; but 41 geometric classes that cover most conceivable representations</i>	Raster Matrix - Spaghetti Chain-node Planar graph Full topology	Raster image Grid Bounded TIN Spaghetti Non-planar network Planar network Full planar graph Basic	Raster Matrix - Spaghetti - Planar graph Planar graph 3D Point set

Table 4.2. Representations (data structures) supported by a selection of geoinformation standards (Albrecht and Kemppainen 1996).

Now there is little communication between these bodies and therefore, differences are occurring in the way in which objects are described and their descriptions. It is therefore necessary for these groups to start working together before it can be expected that other organisations, such as the software companies, choose to implement the ideas. The trouble is that currently software technology is moving faster than the organisations and bodies trying to standardise it. In this way, ideas proposed by standards organisations are commonly obsolete and superseded by the time they are approved.

Without an agreed naming structure to the data then the analysis of data sets requires metadata to describe the contents of the data and conversion tools to convert the naming to the naming standard used by the application. This will not leave geographic information in much of an improved state than the variety of file formats available today. For applications like VHN that are examining the possibility of using full three-dimensional data sets then it is also important that the naming standard can be extended to allow this functionality. From Table 4.2, it can be seen that the current naming standards are concentrating on the two-dimensional representation. The Triangulated Irregular Network (TIN) representation available will allow for the three-dimensional

representation of landscape but not the simultaneous representation of those objects on it.

4.4 Application Object Standards

Various standards are being designed in parallel for how object technologies should be used. These exist from pure high level coding of software, to the user interface of a database system. Currently there seems to be no best standard arising and, as with the spatial operator standards discussed above, co-operation is required between the standards bodies. These standards are looking to allow the access of remote applications and data across the Internet. This sees a move away from the current monolithic systems, towards open systems that can access data and applications from anywhere (Figure 4.2). Presently, the major software developers are still producing monolithic components able to cope with all aspects of GIS.

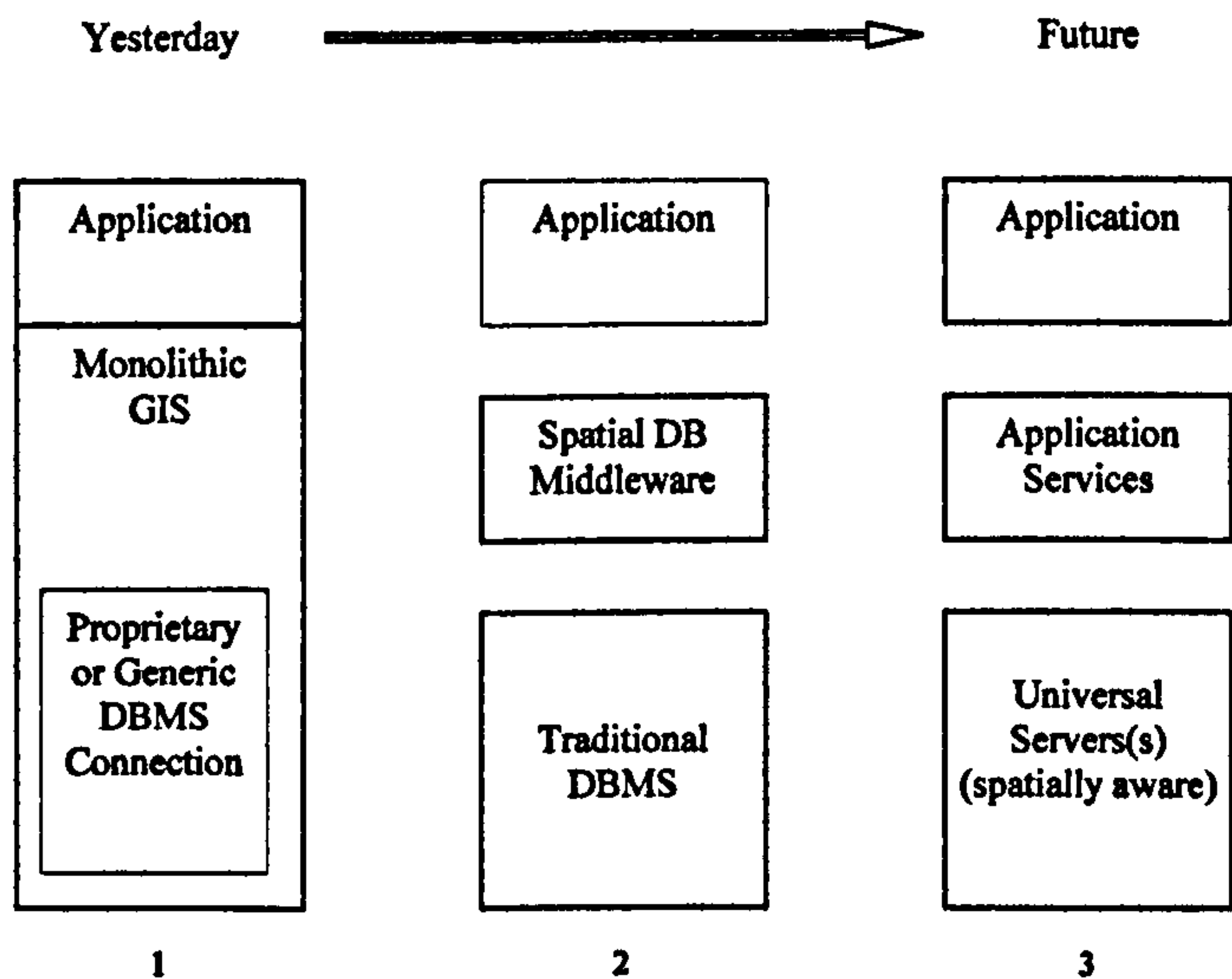


Figure 4.2: The migration from traditional GIS

Figure 4.2. 1) In monolithic systems, which provide many functions, all resources are contained in a single software system or product. 2) Interfaces, even if not entirely open, enable a GIS application to access other database management systems. 3) Open interfaces on a variety of products enable a much more flexible approach. Some geoprocessing services are available in the computing environment, much as standard computer graphics and mouse/window/menu services are available to applications today in standard computing platforms (McKee 1998).

The move to open systems is being enabled by the development of new programming languages and technologies. One of the most widely used object-oriented languages is the C++ language. This was designed to allow the creation of libraries of reusable code, however what is really desired is the ability to create reusable binary objects. Binary components themselves still tend to be designed towards a specific operating system, Windows or Unix for example, or often a specific compiler. The deficiency in C++ to create easily moveable binaries was one of the motivations towards the development of the Java language, though these binary Java objects (JavaBeans) are only available to Java programs (Microsoft 1998a). Therefore, standards are required that specify how a binary object may be accessed, regardless of the operating system. Currently this is happening in two ways. Standards groups are trying to define how objects can operate on any platform, whilst software companies, for example Microsoft, are also looking at getting their operating system on all platforms. It is far better to follow the standardisation route, rather than getting to the stage where all software relies on one operating system.

Figure 4.3 illustrates how an object-oriented approach will work. The various interfaces, facilities, and domains all communicate through a request broker. The different standards bodies (detailed in the following sections) are working on the standardisation of various parts of this architecture.

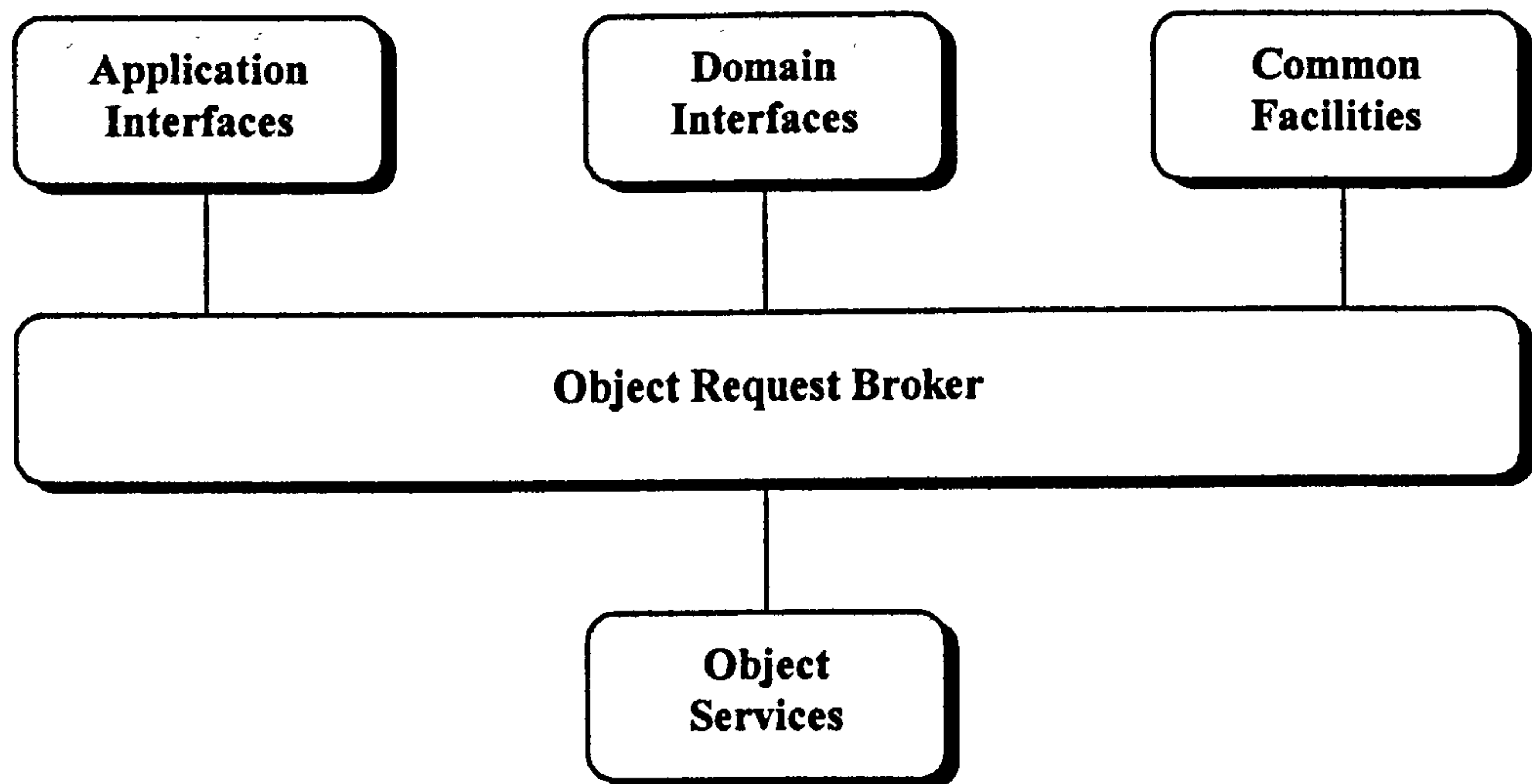


Figure 4.3: Object model architecture

Figure 4.3. In an object-oriented system, a request broker supplies the interaction between the various interfaces, facilities and services. The broker decides which is the best interface, facility, or service to process the event (Schmidt 1998).

Schmidt (1998) describes the various duties of the sections of the object model architecture as follows

- object services are domain independent interfaces that are used by many programs. For example, a service for the discovery of all available services is necessary regardless of application or domain
- common facilities are oriented towards the end-user applications, for example, specifying a document model for facilitating the linking of a spreadsheet to a report
- domain interfaces are oriented towards a specific application domain, for example, telecommunications, medical and geographical domains
- application interfaces are developed specifically for each application, although can include general areas such as input and output of data

The construction of the object model architecture relies on all software moving across to this concept. Unless there is a requirement from industry and users, this will not happen. Many applications have still not moved to encompass the possibilities of today's computing technology, in fact a number of programs only run in DOS.

Applications such as VHN are looking to be able to supply a lightweight application interface in this architecture and, through the ORB, allow the user to interrogate and investigate available data. As stated earlier, this requires a framework in which the component interfaces are clearly defined so that they are able to interact with each other and the user commands. The sections following detail current approaches to the definition of frameworks for individual applications, databases and GIS, as well as more system wide approaches for how applications and the data they use will communicate.

4.4.1 The Object Management Group

The Object Management Group (OMG) is a non-profit consortium dedicated to promoting the theory and practice of object technology for the development of distributed computing systems. OMG was formed to help reduce the complexity, lower the costs, and hasten the introduction of new software applications. The OMG has over 600 members from both business and academia. Some of the members of interest to the GIS community are Oracle, ESRI, Microsoft, and the Open GIS Consortium. The aim of the OMG is to standardise the framework on which object-oriented applications are based (OMG 1998).

The OMG Object Model is based on a small number of basic concepts; objects, operations, types, and subtyping. An object can model any kind of entity, for example a person, a ship, a document, a department, a tuple, a file, a window manager, or a lexical scanner. A basic characteristic of an object is its distinct identity, which is immutable, persists for as long as the object exists, and is independent of the object's properties or behaviour.

Profiles exist to group components. A particular domain will group components that provide extensions that the domain considers important to meet the needs to its specific

user community. Profiles for these domains can be technology-based, for example databases or programming languages, or application-based, for example CAD or Finance.

Current work of the OMG involves the specification of an Object Management Architecture (OMA) which encapsulates a range of fields. The OMA is an architectural framework with supporting detailed interface specifications. The OMA is made up of various components. These are the ORB, the object services, the common facilities, the domain interfaces, and the application objects.

The OMG have also defined an object model for common object semantics when specifying the externally visible characteristics of an object. In this way, applications can be developed to work with applications that possibly have not yet been developed or conceived.

The OMG's main technology is the CORBA (Common Object Request Broker Architecture), with the CORBA 2.0 specification accepted in December 1994. The next major release of this specification, CORBA 3.0, is due towards the end of 1999. CORBA is a means for any application and hardware to interact, independent of who designed them or where they are located. The CORBA specification defines the mechanisms by which objects make requests and receive responses through the ORB. As well as defining the interface definition language and the application programming interface for specific implementations of ORB's. The ORB then provides the framework for interoperability between applications and data, on different machines. CORBA has been postured as being both language-neutral and independent of ORB and CORBA object implementations. CORBA supports bindings for C but it is anticipated that it will eventually support language mappings for C++, Smalltalk, and COBOL.

The work of the OMG with the CORBA interface allows for the creation of more application specific interfaces such as that being done by the Object Data Management Group (Section 4.4.2) and the Open GIS Consortium (Section 4.4.3). The ORB framework defined by the OMG and accessible through the COBRA interface allows for GIS applications to be broken down into task specific applications that interface with each other to produce the current functionality. The framework gives lightweight web based applications, such as VHN, accessibility to the same component applications as the heavyweight GIS applications, allowing them more functionality when required. Furthermore the distributed background to the ORB architecture means that these web applications can access components on other registered servers as easily as those on the host machine.

4.4.2 Object Data Management Group

The Object Data Management Group (ODMG) is a consortium of object-oriented database management system (ODBMS) vendors and interested parties working on standards to allow portability of customer software across ODBMS products. The ODMG 2.0 specification was released in July 1997. This specifies component technology for object-oriented database products and languages. The object query language (OQL) specified by the ODMG 2.0, ensures the portability of applications across platforms, in the same way that SQL works for relational databases. The final goal of the standard is to combine programming languages and database systems into a single environment (Ereno 1998).

Other Sections of ODMG 2.0 allow for the specification of an object definition language (ODL) and techniques to bind to languages such as C++ and Smalltalk. It does not cover all possible areas of functionality; for example, it does not cover

distributed database interoperability or versioning. However, it covers all the basic capabilities necessary for an application to use an ODBMS, to create, modify, and share objects. It is expected that applications written to these interfaces will operate across all compliant ODBMS implementations. Continuing work is planned for later releases that will address added functionality, this will include tracking evolving related standards (e.g., changing C++), and mapping to further languages and domains (ODMG 1998).

The ODMG Object Model is intended to allow portability of applications among object database products. It provides a common model for these products by defining extensions to the OMG object model that support object database requirements. In particular, the ODMG model extends the OMG core to provide for persistent objects, object properties, more specific object types, transactions, and queries.

Like the OMG, the ODMG's members come from both business and academic backgrounds. However, here there is less active participation from the members in the development of the standards. Members on the board are there mainly as a reviewing body. The ODMG submit their work to other relevant bodies such as OMG, ANSI, and ISO to encourage commonality across all standards, a step towards what is required for these standards to succeed.

The approach of using an object-oriented database allows for encapsulation of methods on the data. As suggested earlier in this chapter, this allows for such methods as data validation and combination to be stored with the data rather than the application. This transferral of intelligence from the application to the data allows the applications to become simpler and lightweight. It also allows new applications to be able to process and display the data with no knowledge of its format or contents.

4.4.3 The Open GIS Consortium

The Open GIS Consortium, Inc. (OGC) is dedicated towards developing open system approaches for geoprocessing. By means of its consensus building and technology development activities, OGC has had a significant impact on the geodata standards community. The OGC has actively promoted open GIS to enable the integration of geoprocessing, with distributed architectures that define the worldwide infrastructure for information management. In emphasising open system approaches, OGC works to facilitate the use and acceptance of advanced information processing technologies in the geosciences. This is to promote the integration of geoprocessing techniques with industry standard tools, for the development of more effective, decision support capabilities (OGC 1998).

The direction of the OGC is set by a board of directors, selected to represent essential constituencies in the geoprocessing community. The OGC board represents both public and private sector users interested in finding more integrated and effective ways to use geographical information, for example, to support problem solving in such areas as environmental monitoring and sustainment, transportation, resource management, global mapping, agricultural productivity, crisis management and national defence.

Amongst its membership, the OGC has a range of backgrounds including Autodesk, Oracle, ESRI, MapInfo, OMG, and Smallworld. The main aim of the OGC is to produce a single model for all spatio-temporal applications. Along with this will be specifications for the major database languages and computing environments, so that they can connect to the open geodata interoperability specification (OGIS) framework. The two main areas of development are in the OGIS geodata model (OGM) and the OGIS reference model (ORM).

The OGM is the core component of the framework. It consists of a hierarchical class library of geographic information data types that comprise the shared data environment and unified data programming interface for applications. Cohesive, comprehensive, and extensible, encompassing all current geospatial and temporal descriptions of data, presentation issues, analysis modes, and storage and communication issues, the OGM will be the language of OGIS. The OGM will be the interface to geodata transfer standards such as the Spatial Data Transfer Standard (SDTS), SAIF, and Digital Geographic Standard (DIGEST). It will be interoperable with SQL3-MM, the next generation of SQL, which will be object oriented and which will support multimedia entities, including geospatial data.

The OGM will support object queries, read/write of multiple formats, temporal modelling and long duration transactions. It will, to accomplish these objectives, include sophisticated GIS definitions of spatial objects, fields, and functions. The ORM describes a consistent open development environment characterised by a reusable object code base and a set of services. The design approach for the OGM determines the set of services that must be supported in the ORM. Therefore, the ORM requires directories of services and databases, which will support complex query processing. It also specifies standard methods for requesting and delivering geospatial transformations and processing tasks. The ORM will also facilitate transformations between private data and OGM constructs, as well as co-ordinate conversion and raster/vector conversion. It also manages visualisation and display, and it supports data acquisition (OGC 1998).

McKee (1998) suggests that membership of the OGC enables technology providers to

- share the cost of technology development
- ensure the OpenGIS Specification meets their product needs
- begin development of OpenGIS conformant products
- develop relationships with potential customers in the consortium
- solve shared problems
- gain perspective on global strategies and opportunities through the involvement in OGC of key European and Asia/Pacific academic, industrial and government scientists

Data transfer standards are a start towards open GIS, but these alone are not enough.

What is required is a means to access data from another system without the need to convert files off-line, enabling them to be opened seamlessly from within another application. Current systems that can do this require knowledge of other applications' internal file formats. The aim of the OGC is to design a specification that says how data files should respond to requests for information about the data that they contain. The internal structure is still proprietary to the application. What is standardised is the external view of this data.

4.4.4 Microsoft

One of the major suppliers of operating systems, Microsoft has an ability to implement ideas without the need for standards or the approval of other bodies. This puts them, as an organisation, in a strong position to influence the development of standards and the direction of computing. If the core operating system functionality changes, then eventually the software that runs, and is built on that technology will change. Microsoft

has a history of developing software that will allow for the exchange of information between applications, implementing ideas for sharing both application source and data. These include dynamic link libraries (DLLs), the component object model (COM), object linking and embedding (OLE) now renamed ActiveX, open database connectivity (ODBC) and dynamic data exchange (DDE).

COM is Microsoft's architecture for finding and running binary software components. Three types of components are supported; in-process (a DLL), local (another application), and remote (a DLL or application on a different system). The structure defined by COM provides the basis for interoperability between components written in different languages on different machines, and is an alternative to CORBA technology. ActiveX is designed to support interaction between client and server applications and is built on COM. At the moment ActiveX and COM are available on Microsoft Windows and has been ported to some Unix platforms, though interaction between ActiveX and COM objects with CORBA objects is being developed (Ereno 1998).

Currently, the main applications that use ActiveX technology are Internet Explorer and the Microsoft Office package. Although mainly available on the Microsoft platform, there are developments for Apple Macintosh and UNIX, especially with the underlying COM technology that ActiveX is built on. ActiveX allows for the remote implementation of applications written in a language such as, C++, VBScript, Java, ECMAScript and database development languages. It is not seen as a replacement for languages such as Java, but as an extension to their capability. (Microsoft already implements Java in their browser as an ActiveX control). This technology is beginning to allow anyone the ability to easily and quickly create an application using the Internet.

OLE 2 provides an application integration framework for Microsoft Windows. OLE 2 defines the Component Object Model, which specifies a programming-language-independent binary standard for object implementations (i.e., it specifies what the implementation of the objects has to look like). Any object conforming to this standard is a legitimate Windows Object, no matter what language is used to implement it. The programming model is synchronous, based on a lightweight remote procedure call (lightweight because, at least now, the calls are not remote; they are all made on one machine).

For the sharing of data, the model used is the Universal Data Access architecture, as detailed in Figure 4.4 below. This allows for the sharing of data regardless of where or how it is stored (Kite 1998). This illustrates the relationships between the current component technologies supplied by Microsoft.

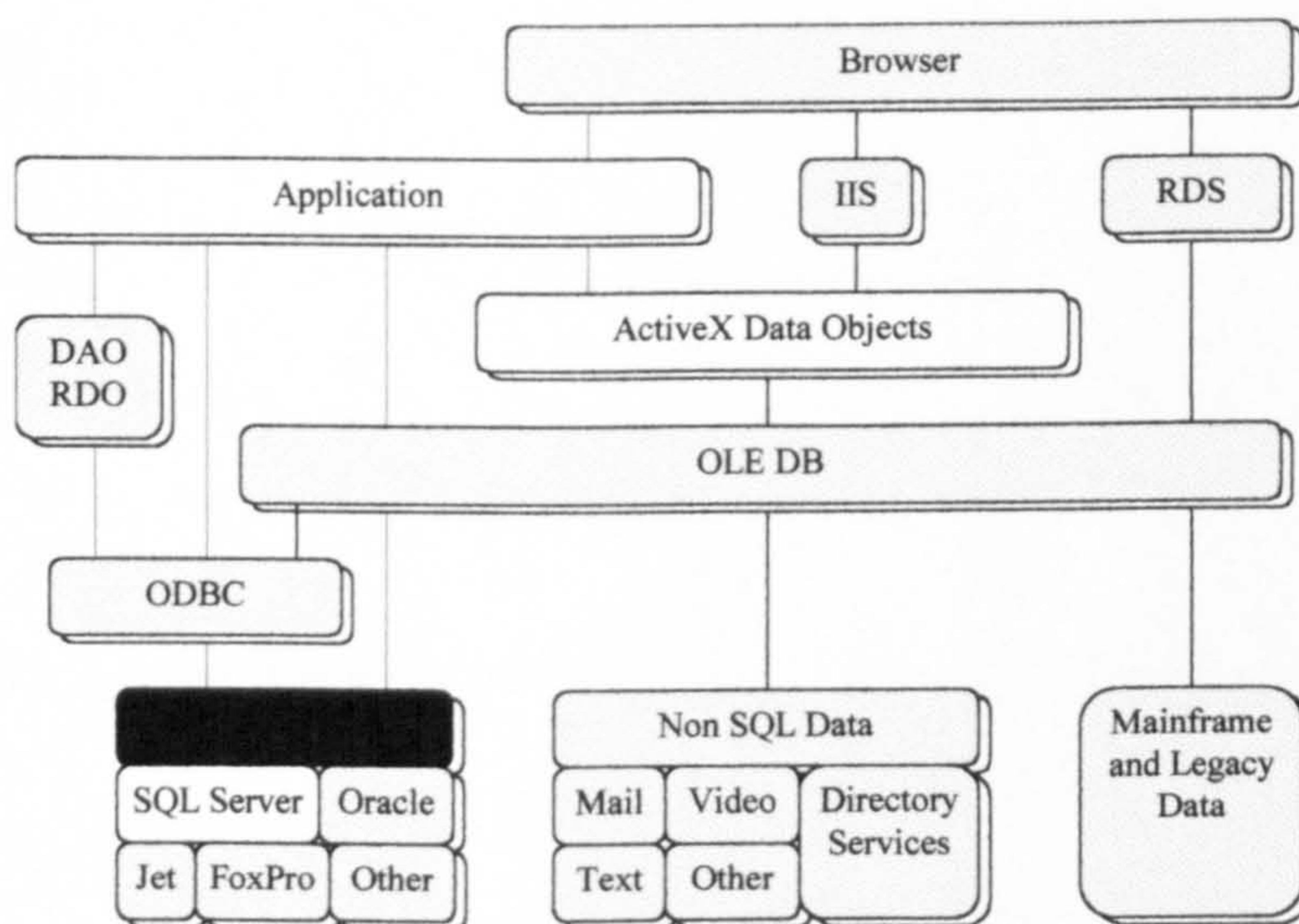


Figure 4.4: The Universal Data Access architecture

Figure 4.4. Diagram illustrating how the core technologies from Microsoft fit into the process of universal access to various data stores either local or networked (Microsoft 1998b).

The Internet Information Server (IIS) and the Remote Data Service (RDS) are technologies for distributing data and accessing databases over the Internet based on

web server technology. These are the main means to implement the ideas of distributed software and information as discussed in this research. These services allow requests for information to be made from a client application on a remote system to local data on the server.

In the development of their data access model Microsoft have surpassed many of their original technologies for accessing data. However, for the benefits of compatibility with older applications these are still included. The original technology was Open Database Connectivity (ODBC) which allowed the usage of standard function calls to access any database for which an ODBC interface existed. This included standard relational database systems as well as spreadsheets and text files. Once connected to the database queries could be performed by the use of the SQL. ActiveX Data Objects (ADO) is the successor to the Data Access Objects (DAO) and Remote Data Objects (RDO) COM technologies. It offers a much higher level application programming interface (API) than that supplied by ODBC and follows a more object-oriented approach. A Java implementation (ADO-for-Java) allows for system independent usage of the technology (Ereno 1998).

Microsoft has recently rebranded their open system technology under the umbrella title of Windows Distributed interNet Applications (Windows DNA) Architecture. The core elements of the architecture are exposed to each other through the idea of COM. Microsoft (1998c) state that Windows DNA consists of

- presentation services (HTML, DHTML, scripting, components, Win32 API)
- application services (IIS, MSMQ, MTS, COM+)
- data services (ADO, OLE DB)

- system services (directory, security, management, networking and communications)

The aim of Windows DNA is to integrate a web-based interface to all data and applications, something already done in the Microsoft Money product. Here the entire user interface is built in Dynamic HTML. The main differences between the technologies available now and those in Windows DNA are the level of programming and the targeting towards Internet based applications. Windows DNA presents a higher level of abstraction to the coding than that currently supplied by technologies such as COM and OLE.

The architecture shown in Figure 4.4 closely represents that for the prototyped VHN application. This is not just through the VHN being based currently around Microsoft products, but the architecture makes a distinct separation between the interface, applications and data. Keeping these three components of the project separate allows the user the most freedom in what they are able to do. Applications can be added, or removed, as their capabilities are required, as can the various data stores that these applications interrogate. Removing the interface from the underlying application means that the interface can be tailored to be as complicated or simple as the current project or user requires. What is required from the Microsoft approach is the ability to communicate with systems running on other platforms such as Unix or Linux. Without this capability then restrictions are imposed on the project through the framework rather than through the availability of applications or data.

4.5 Querying Data

If the format for the storage and exchange of data is fixed, the programming language must be capable of extracting a selection of the data required from the set. The current language for querying data is SQL. This is designed for usage with relational databases

and therefore does not understand the concepts of object-orientation or allow spatial searches.

The two committees who are jointly developing SQL are ANSI (X3H2) and ISO (ISO/IEC JTC1/SC21/WG3) who have for some time been adding features to the SQL specification to support object-oriented data management. This extension to SQL is often referred to as SQL3 and includes the capability to support user-defined abstract data types (ADT's), including methods, object identifiers, subtypes and inheritance, polymorphism, and integration with external languages.

One of the basic ideas behind the object extensions is that, in addition to the normal built-in types defined by SQL, ADT's may also be defined. These types may be used in the same way as built-in types. For example, columns in relational tables may be defined as taking values of user-defined types, as well as built-in types. ANSI (1998) states that an ADT is defined by

- specifying a set of declarations of the stored attributes that represent the value of the ADT
- the operations that define the equality and ordering relationships of the ADT
- and the operations and derived attributes that represent the behaviour of the ADT, implemented by procedures called routines

The only persistent way of currently storing an ADT is by use of columns in a relational database structure. SQL3 is still mainly based on a relational database framework of tables, rows, and columns. The object query language (OQL) as defined by the ODMG allows for full use of object oriented database systems. SQL3 object extensions are also being pursued through extensions to the table facilities in SQL. The new row identifier

facility allows the DBMS to maintain a unique identifier for each row, and allows applications to make use of this identifier. A row identifier can also be used as a column value and/or foreign key. The facility is specifically intended to help reduce the difference between SQL3 rows and objects in conventional object models.

Enhancements have also been made to the facilities for defining tables (relations) in SQL3, including row types and row identifiers, and an inheritance mechanism. Additional facilities include control structures and parameterised types to make SQL a computationally complete language for creating, managing, and querying persistent objects. The added facilities are intended to be upward compatible with the current SQL standard, SQL92. Using these facilities, routines can be associated with tables to implement object-like operations on rows, and more specialised routines can be associated with subtables to support polymorphism for those operations.

ADT's enable the definition of data types similar to the idea of objects. ADT's create a 'blackbox' approach to displaying and utilising multimedia data of differing formats. This produces for the user a list of media types and related operations that can be carried out on them. This 'box' can encapsulate such operations as editing, displaying, and querying.

The current SQL3 specification includes the ability to create user-defined abstract data types. SQL3 ADT's allow for the inclusion of methods, object identifiers, subtypes and inheritance, polymorphism, and integration with external languages. Object enhancements are also being added to both the type and table definition facilities of SQL. User defined ADT work in addition to, and combined with, the standard built-in types of SQL. An ADT definition encapsulates both the attributes and operations of an object in a single entity.

4.6 Summary

Recently many applications have been designed as objects which users can program into their own applications. Now it is more common to include these objects with the application when it is distributed rather than to assume the user has the object. Applications are also tied to specific objects: it is not yet possible to just send out a command to load a map sheet and rely on the system to decide which object is best suited to the task. Microsoft has gone some way to implementing this idea in Windows but it relies on file extensions being registered to objects. This is not always feasible and little safeguard is built in for generic operations if no specific object exists to cope with an extension.

An example of a commercial GIS application, which has been built as an object expected to be implemented into other user applications, is MapObjects from ESRI. This is still a very monolithic object with users only building an interface to the object. Instead of there being separate objects for graphical and textual data handling, the object tries to handle it all. Here a purely graphical application is bloated by the inclusion of database access and control software which is not required. There is still a long way to go before specific objects are written to handle a single task.

As shown in the prototype interface to VHN (Chapter 3) the use of components can enhance the user interface to an application with little overhead to the developer. In the VHN example, the three-dimensional visualisation component and the talking wizard component are interfaced to each other via simple scripts to present a tour of the townscape to a user. This approach allows the developer to concentrate on developing the topic specific parts of the application while using existing components for generic areas.

This chapter has shown that component technologies exist commercially although they are not currently fully developed or integrated, especially in the usage of geographic information. This can be attributed to two reasons: the development has been carried out by bodies outside the area of geomatics and people are still waiting for the formalisation of standards from ISO and CEN (Section 4.2).

Chapter 5 leads on from this to discuss current techniques for storing geographic data. To allow for the move to data being stored remotely, to avoid it being duplicated and becoming out-of-date, means that it is necessary to consider how data is structured. It is important to remember that the external representation of data through its interface does not have to correspond to the internal structure, and that some forms of data may have multiple interfaces. The chapter details structures developed as part of the research, as well as those used in other areas of research that can be applied to geographic data.

5. Data Structure

Nature exhibits mathematical structures to the objects it contains. Snowflakes exhibit symmetry and always have six sides, clouds form in fractal patterns, and for the majority of plants, the number of petals falls in the Fibonacci sequence (Stewart 1995).

The structure of data used to model the world, and hence Nature, in a computer rarely uses the concepts Nature itself uses. Here, all too often an order is enforced on a data set that is not natural of the data itself. The world exhibits a hierarchical object-oriented structure, especially in how it is interpreted by humans. For example, a tree is made up of a trunk, leaves, and branches. In turn both the trunk and branches are made up of wood, sap, and bark. The human conception of the World groups objects together to form the whole, it is very rare in everyday life to refer to objects by their chemical compound. However, when stored in a computer the tree would be reduced to being only the lines that convey it in a graphical form. The structure of computer data can be influenced by many requirements

- how compact does it need to be
- does it need to be human readable
- will there mainly be sequential or random access
- is it designed to be displayed on its own or with other files
- is the data graphical, textual, or aural
- does the data need intelligence or just a visual impact

A combination of these requirements defines how the data is finally stored. By crossing the boundaries and examining ideas outside the area of Geomatics and current research, it is possible to propose new ways to structure geographic data. The following sections

examine the storage of graphical data, but do not examine in depth textual and aural data. Many of the concepts though, can be applied to these two types of data. For the purposes of the VHN project, the main data structure requirement was for a three-dimensional model with related two-dimensional images. With relation to the requirements on data structures mentioned above, both of these structures were required to be

- compact to allow fast transmission across the Internet,
- allow random as well as sequential access to allow the user to browse or follow set routes,
- contain intelligence to allow the various types of information to be linked together.

This chapter examines the concepts and difficulties of storing data in two, three and four dimensions and the relationship between them. The structure of the data not only affects the physical storage of the data, but also where and how that data is stored and accessed later. Into this comes the problem of how to name the objects contained in the data, as mentioned earlier in Chapter 4.

5.1 Two-dimensional Data Structures

In GIS, there are two main forms of two-dimensional data structure, raster, and vector. Raster represents a co-ordinated array of values, which are commonly displayed as colours, to store information about an area. Vector data, at its simplest, is a list of ordered pairs of co-ordinates that, when joined, represent the area.

However, it is only necessary to examine the list of file formats available to realise that the two data structures are not simple. The ranges of formats exhibit the ability to store

extra information, rather than just the visual information. These differences come from the compression algorithms used to reduce the size of the data, the encoding of layers into the data, the storing of multiple images or geo-referencing.

The VHN project uses two-dimensional data formats to store the vector maps that allow for faster navigation around the virtual world and to store raster photographs and images of the area. The following sections describe formats that can be used to store these data sets and the development of a compressed vector format for map data.

5.1.1 Two-dimensional Raster formats

As stated above, raster data consists of a regular array of values that is continuous across the bounds of the data. In this basic structure the volume of data is defined by the resolution and size of the array and can often be very large. Raster formats exist that allow for the compression of data, in applications where file size can become a problem. Most of these compression techniques exist in the public domain; for example, the joint photographic experts group (JPEG) and tagged image file format (TIFF) formats. Others through their use have come into the public domain, although companies hold patents on the compression algorithms used, for example the Lempel-Ziv-Welch (LZW) algorithm used in the graphics interchange format (GIF) owned by Unisys. Recently, Unisys has made a bid to recoup some of its lost royalties from individuals producing and using GIF's on the Internet (Seife 1999). This distinction between companies having public domain and private file formats exists in vector data as well; for example, Autodesk's public DXF, and private DWG formats.

For the purpose of the VHN project the JPEG format was used, as this supplies a good compression ratio with minimal impact on the display of the image. The JPEG format is becoming more popular for web pages (in part due to the Unisys royalties claim), and

the majority of web browsers have in-built capabilities for displaying the format. Some of the newer formats, for example the Portable Network Graphics (PNG) format as proposed by the W3C, require further plugin components to be installed on the host machine. As these new formats become more accepted in the web community and as support for them grows in commercial graphics packages, their use in web pages will increase.

An enhancement to image format that would aid the VHN project and many other web based applications would be the availability of storing a thumbnail image in the same file. The VHN project displays thumbnails on the main page and generates hyperlinks that allow the larger image to be opened in a separate browser window. Thumbnail images are reduced quality images usually created by reducing the dimensions of the image, and allow a web page to display a preview before the user has to download the large high quality image. Currently, thumbnail images are created separately and therefore can become out of date if they represent changing data.

5.1.2 Example Binary Vector Format

As part of the research, software was developed to allow for the fast viewing and compact storage of vector NTF level 1 data, such as that used for the Landline product from OS. However, the data format was designed to be generic enough to allow conversions from other vector formats. Although not initially designed as a component that could be included in a web page (or other software) there is no reason why the code can not be implemented in this way.

NTF itself is a human readable ASCII format, which means it contains a lot more information in the file than is required for a computer to be able to display the information on the screen. The binary compression of the NTF file represented here

retains all the graphical data of the NTF file; some of the topological and header data of the file can be discarded for the compact format. For a complete application then there should be the possibility of retaining and utilising all the original data available in a NTF level 1 file, as well as that available in the other levels of NTF.

Data for transfer over networks such as the Internet, or for field use, is required to be as compact as possible to keep costs down. When accessing files via the Internet rather than from the hard disk of a local machine the display time can be affected by the time it takes to retrieve the data file from the information server. Therefore, the more compact the file the less of an influence this has on the efficiency of the data. Compact files can save money in the time taken to obtain them from the Internet and in the storage space required for them on a physical medium.

Table 5.1 shows the complete structure for the binary format, SPF, defined for the purpose. The compression was initially designed to fit two criteria, firstly to be small enough to fit maximum data on a CDROM, as well as fast enough to allow quick display and interrogation of the data. The principle behind the compression used here is that for ASCII to convey a four-digit feature code four bytes are required, one for each character that is stored. By using a binary representation, then two bytes can be used to store the same number, the maximum number able to be represented by just two bytes being 65535, or $(2^{16}-1)$. Similarly, six bytes (three for Easting and three for Northing) can represent any OS National Grid co-ordinate, accurate to 1m, in binary, although this could require up to thirteen bytes in ASCII.

Table 5.1: SPF data format

Section	Bytes	Meaning
Header	3	Left
	3	Bottom
	3	Width
	3	Height
Layers	1	Layer Number
	1	Line Style
	1	Line Width
	1	Red Value
	1	Green Value
	1	Blue Value
	1	Layer Name Length (n)
	n	Layer Name Text
Polylines	1	Layer Number
	1	Number of Co-ordinate Pairs (n)
	n*6	Easting and Northing
Symbols	1	Layer Number
	1	Symbol Height
	3	Easting
	3	Northing
	1	Symbol Number
Annotation	1	Layer Number
	1	Text Height
	1	Text Rotation
	3	Easting
	3	Northing
	1	Annotation Length (n)
	n	Annotation Text

Value Ranges	1 byte	0 to 255
	2 bytes	0 to 65,535
	3 bytes	0 to 16,777,215

Table 5.1. The SPF data format is a compact binary format that can contain all the information to display a NTF file.

The SPF format developed has now been implemented in a commercial package, MapAgent from Quest. This system, used by North of Scotland Water (NoSW), allows for the rapid display of OS and NoSW data for engineers in the field. Initial trials of the software show that the SPF format produces an average reduction of 75% to a NTF file. This means that the engineers can carry out far more data into the field. The system has been implemented so that the software and the data are stored on CD's indexed by

areas. Therefore, engineers can just be supplied with a CD for the region in which they operate.

Compressed vector formats like SPF are also useful for the inclusion of map data in web pages, such as VHN. Section 3.6 detailed the vector graphics languages being developed for the web, but as with NTF these are designed to be human readable to some extent and encoded in ASCII. With the large data sets required for geographic information, the amount of data gets unwieldy and there is little requirement for the data to be readable or hand encoded. By compressing the data, this reduces the storage space and the transferral time across the Internet. As mentioned in Section 4.3, if the data format contains an interface that conforms to a recognised standard then the internal structure of the data becomes independent.

5.1.3 Fractals

The concept of fractals is that an object contains smaller images of itself. Therefore, as you zoom into a fractal object it replicates itself. The best-known fractal objects are the Mandelbrot and Julia sets. As mentioned earlier, in nature clouds exhibit a fractal form. It is very hard from the ground to judge the relative size and distance of a cloud. At a distance, the edge of a cloud looks the same as it would look if you were very close to it.

Figure 5.1 illustrates the generation of a fractal pattern. The stages are; the line in (a) is divided by three, by the addition of the triangle, to give (b), this pattern then continues as much as the user requires, building up a higher resolution image. This illustrates geometric self-similarity, where there is a strict equality between the large and small scales. This idea, though not found in natural geographic features, is used in GIS. For example, the Morton order for Quadrees has the same pattern replicated at every level.

A form of fractals conforming to a more natural pattern is statistical self-similarity. Here, the equality between levels is expressed in terms of probability distributions (NCGIA 1990b).

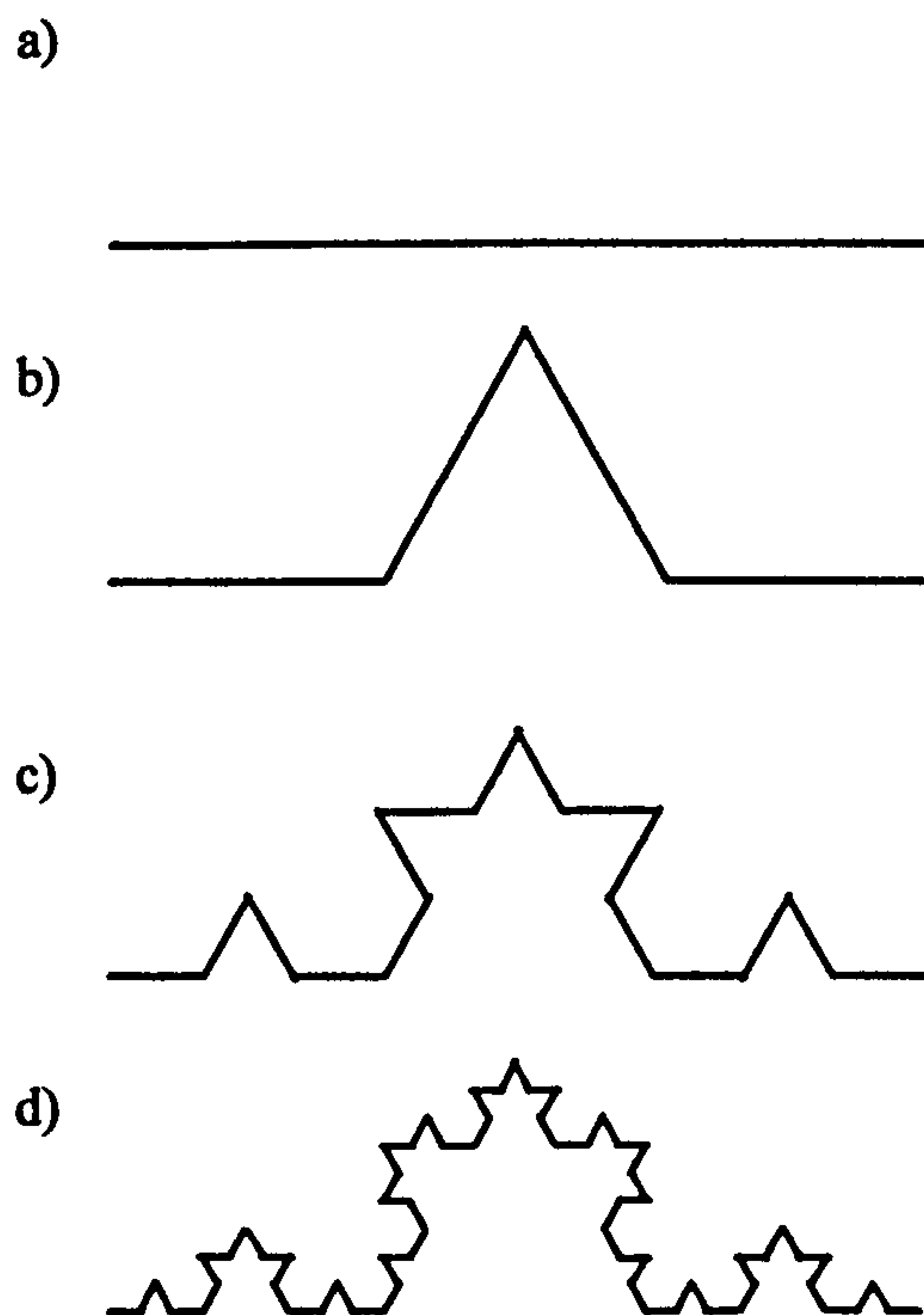


Figure 5.1: The enhancement of a shape using fractals

Figure 5.1. A shape can be enhanced by use of fractal techniques to enhance the detail of the object. At a small-scale (a) is used to represent the shape. As the user zooms into the shape the resolution is increased until eventually (d) is used to represent the shape.

This idea of statistical self-similarity of fractals is useful in modelling natural objects such as coastlines and rivers. A geographical feature can be considered self-similar if it is impossible to determine its scale without any other visual clues. The data can then be structured such that only a relatively low outline has to be stored. Then, as the user changes the scale of the drawing, the outline is either enhanced or generalised by the use of fractal mathematics and probability distributions. The usage of fractals explained here is for two-dimensional objects, though there is no reason why the concept cannot be applied to higher dimensional objects.

The argument against the increased use of fractals for geographic information is that the accuracy of the model is diminished. The counter effect of this is that the visual impact for the user is increased. The final usage of the data has to be considered before fractals are used in this way.

Few applications currently use fractal approaches to the generation of data for display. Where applications can benefit is from the inherent natural look that fractal geometry can produce. In virtual townscape applications such as VHN then the environment can be softened by the use of fractals to represent and enhance natural features, for example trees and rivers. This gives the appearance to the user of a more natural environment as it is no longer made of straight edges, but without the overhead of the data required to represent the detail. All that is required is to store extra information on the object that describes how the fractal pattern will be created.

5.2 Three-dimensional Data Structures

Standards for storing geographical data currently concentrate on handling two-dimensional data, or at the most two-and-a-half-dimensional data. In contrast, standards exist for the storage of CAD and VR data that can handle full three-dimensional data as well as further properties about that data. The extension from two-dimensional data to three-dimensional data increases the volume of the data and the complexity of the topological relationships.

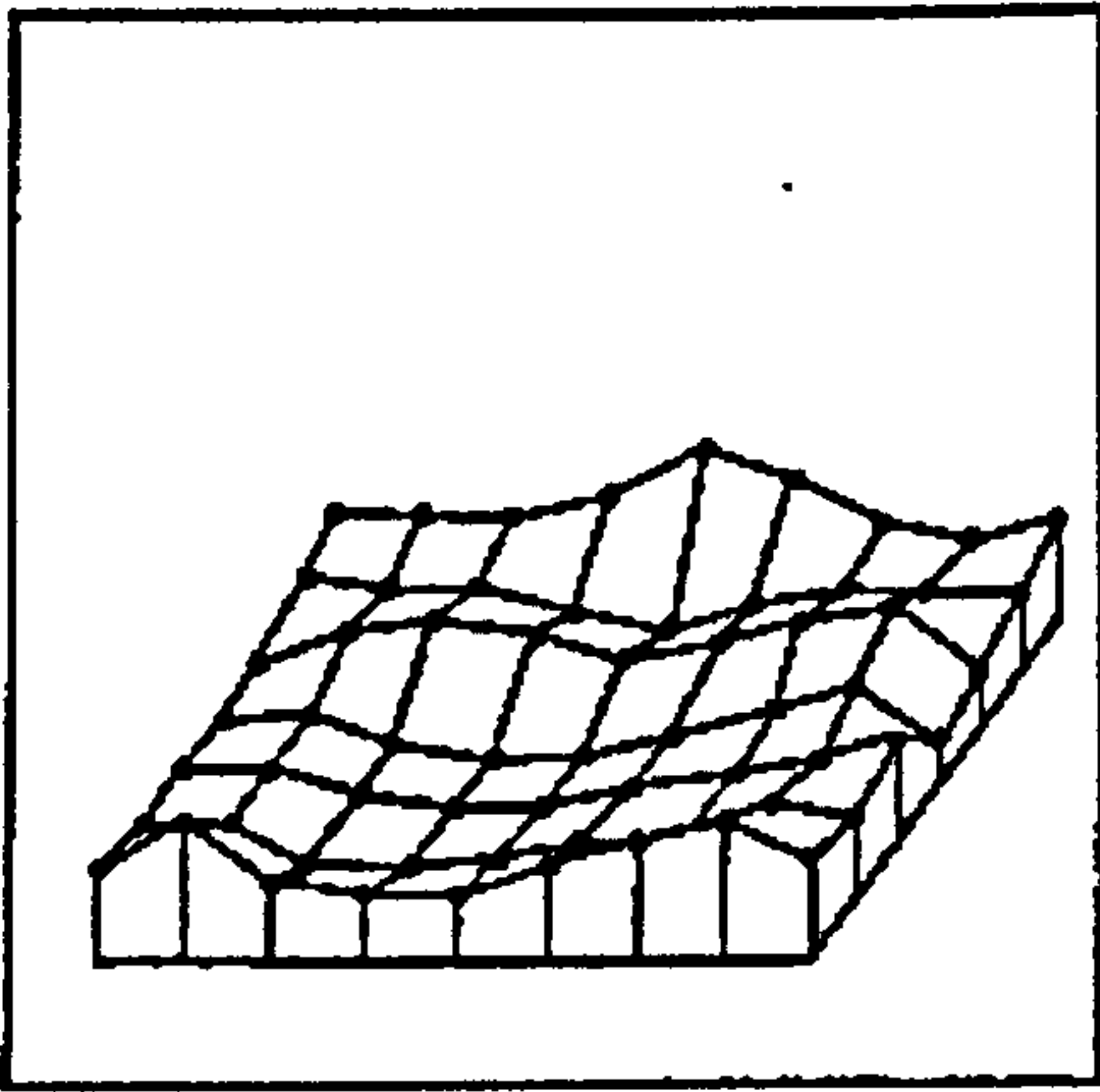
For applications such as VHN, there is the requirement to store both two-and-a-half-dimensional and three-dimensional data. The actual landscape can be stored as a two-and-a-half-dimensional representation with the three-dimensional buildings placed on it. However, it is better to have the option of storing landscape as a three-dimensional data

set for the occasions where the surface does overlap for example around cliffs and caves.

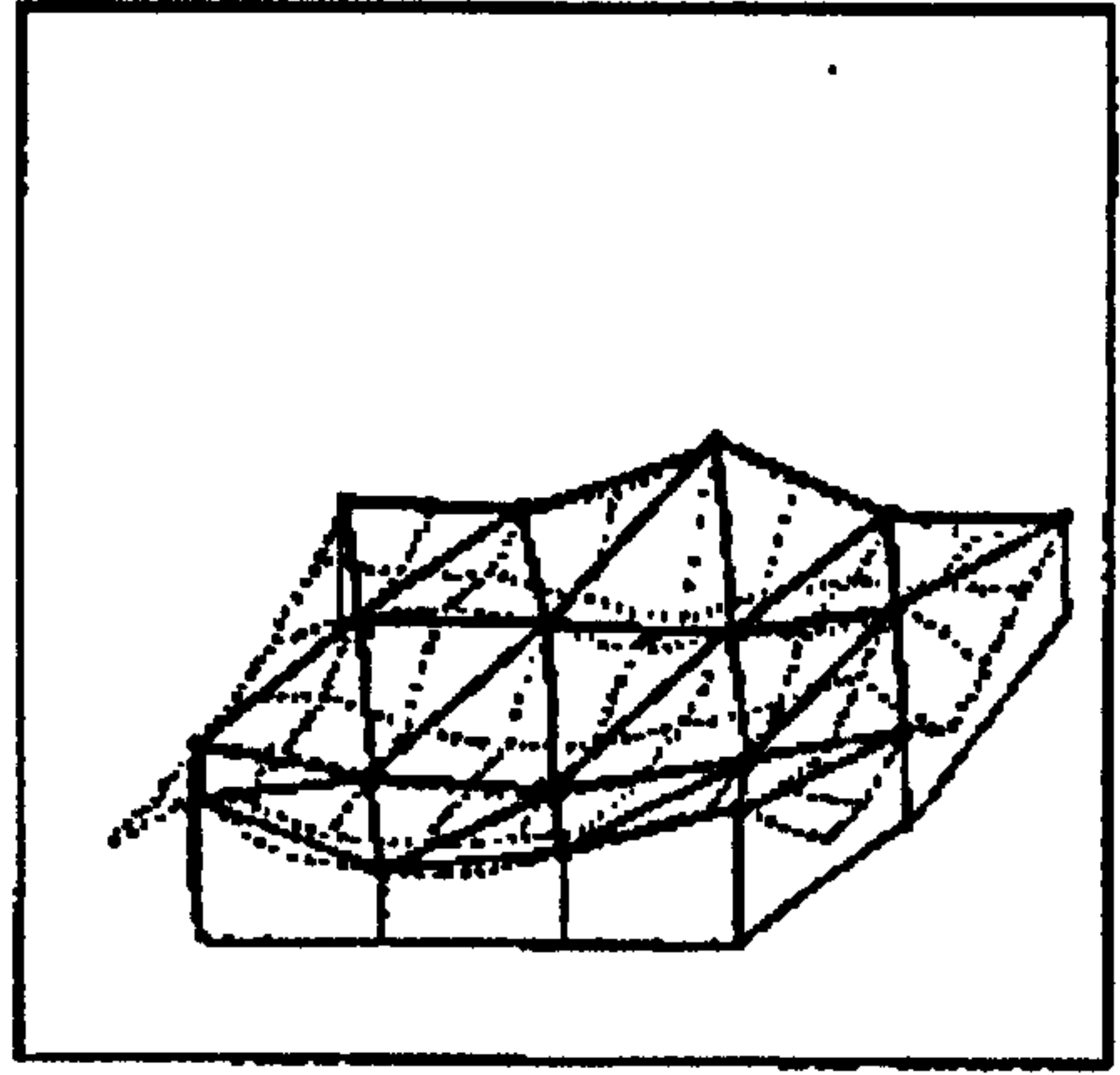
5.2.1 Vector Structures

Many in the geographical community have grown happy with the concept of only formalising the flat world, yet for many applications this concept does not exist. The main current way for representing the three-dimensional nature of a landscape on a two-dimensional map is by use of contours. In a computer representation of a landscape, often referred to as a digital terrain model (DTM), the data can be stored in a number of ways. These are still in the form of a two-and-a-half-dimensional data set, as opposed to full three-dimensional models. Fairbairn (1987) discusses the different data structures as defined by Peuker (1980) that can be used to define a surface.

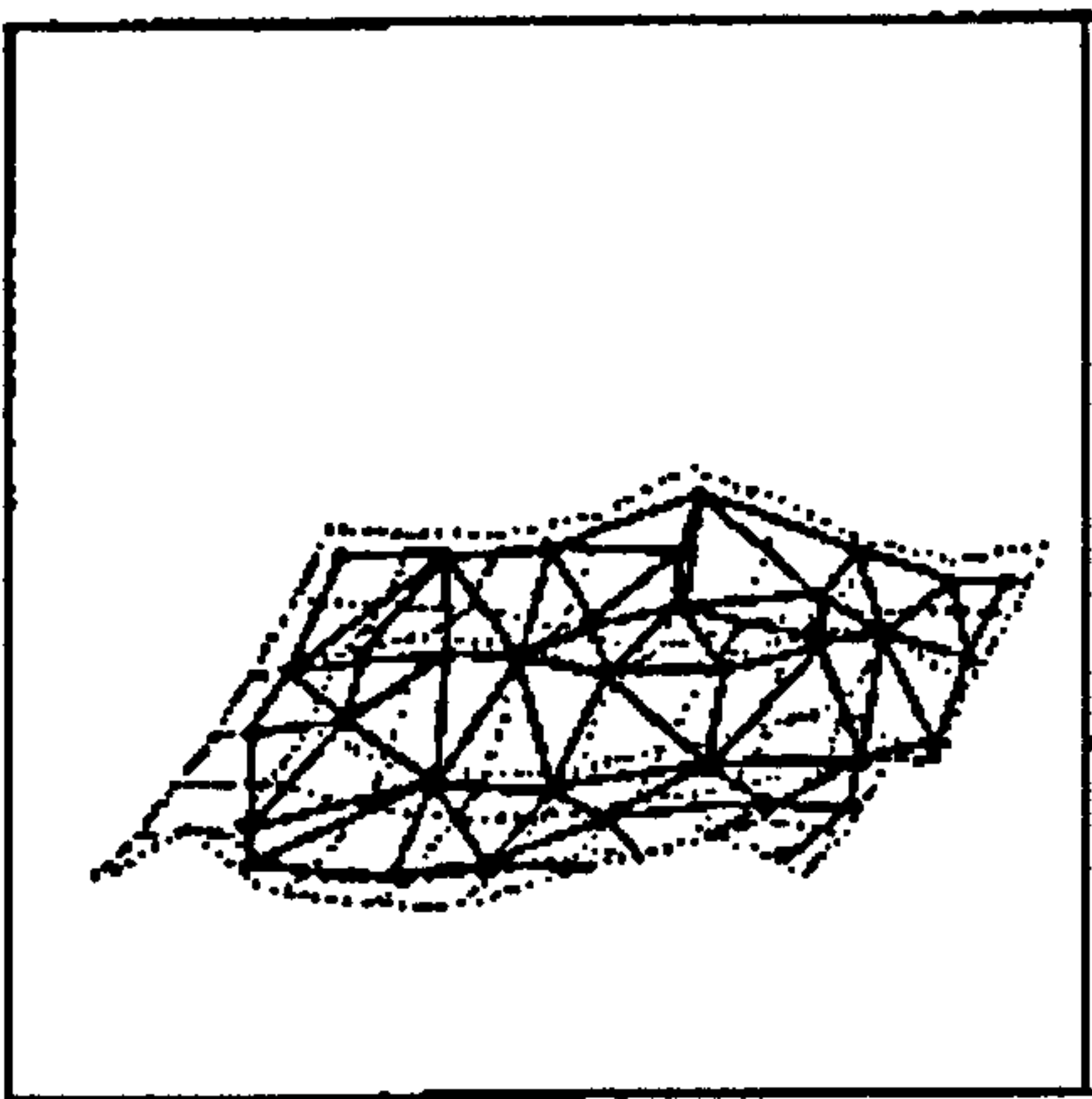
The different approaches to structuring two-and-a-half-dimensional data, as illustrated in Figure 5.2 by Peuker (1980), can depend on the format the data was originally collected in and the uses for it later. A regular grid of points (1) illustrates a systematic approach to the sampling of the area. This arrangement is also the basic format for conversion to a raster based data structure for three-dimensional data. Unless the data is originally surveyed for this structure, then the data has to be manipulated and approximations for the height at the grid points calculated. For increased storage capabilities, this format can be enhanced to that shown in (6). Here area patches are used to allow a hierarchical quadtree structure to be applied to the data.



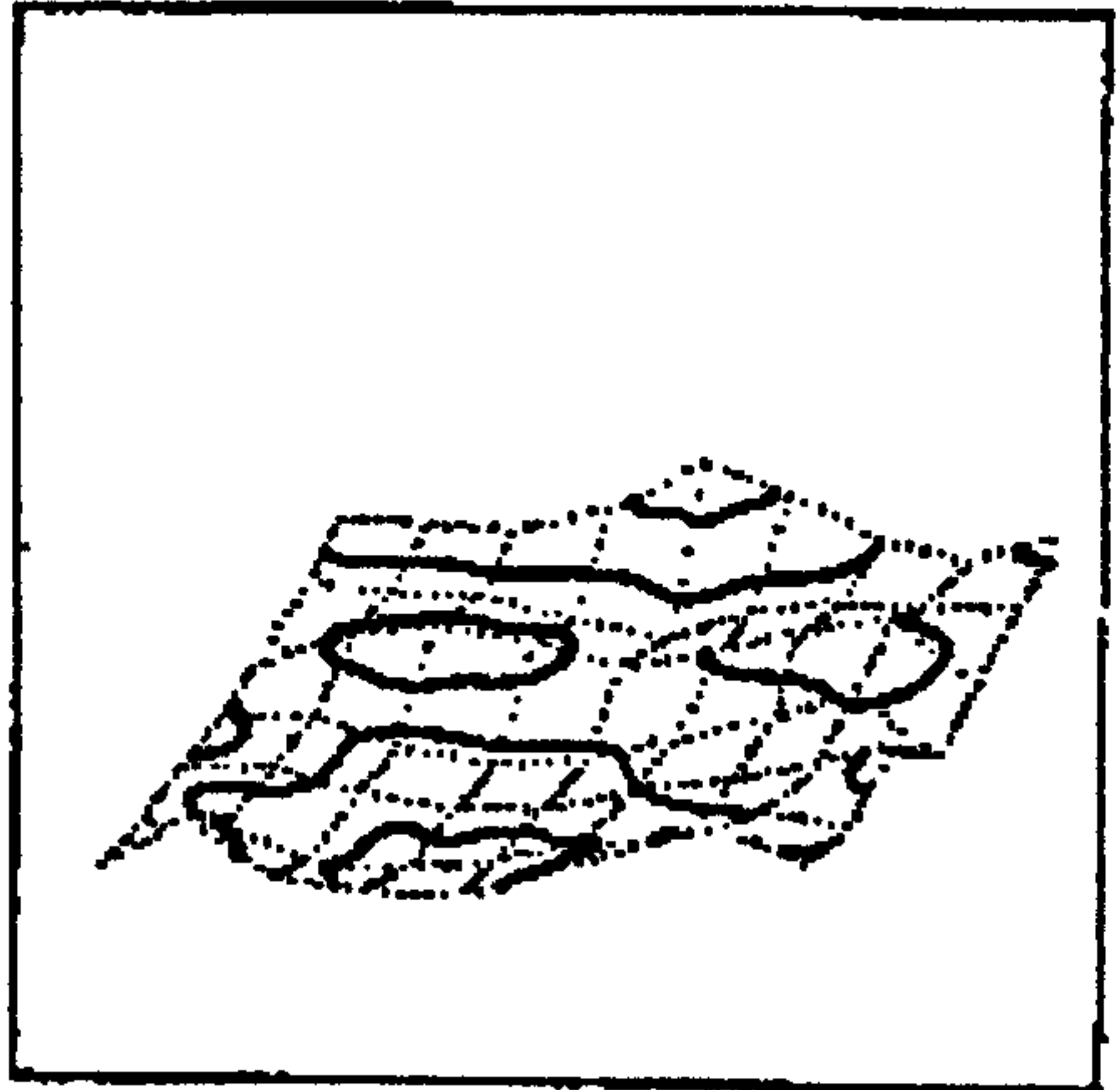
1) Regular rectangular points grid



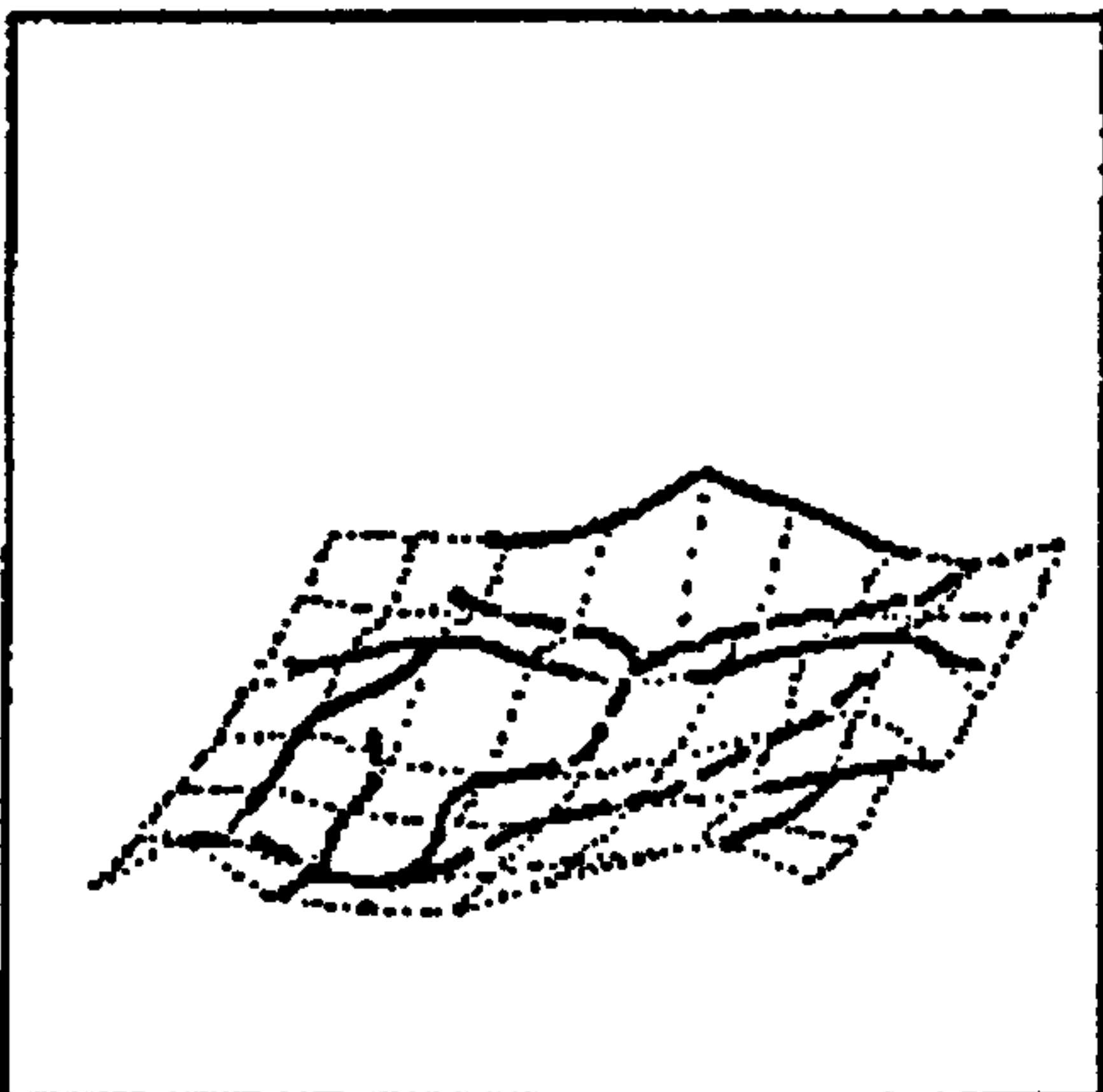
2) Regular triangular points grid



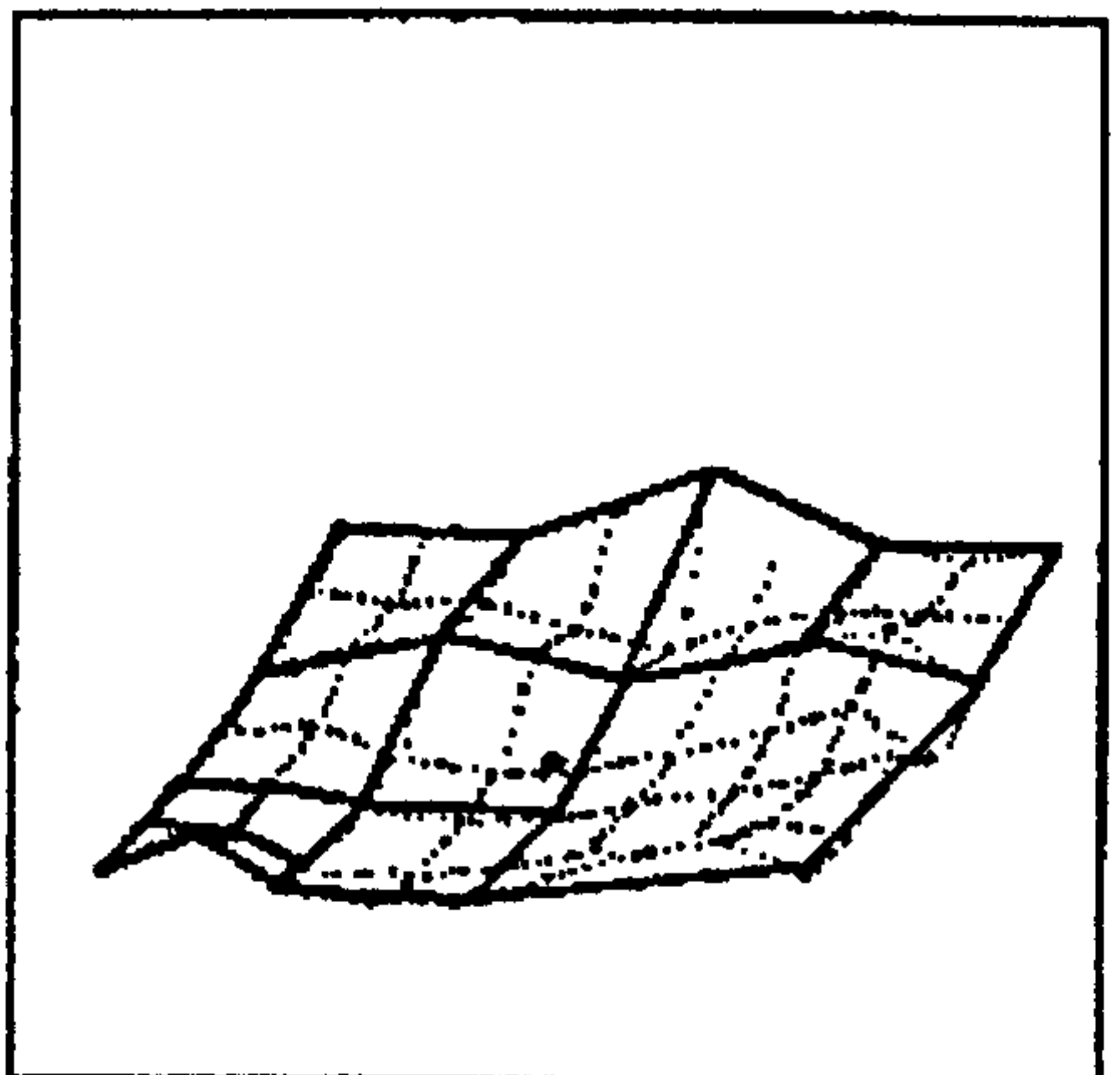
3) Irregular points



4) Contours



5) Three-dimensional strings, including profiles



6) Areal patches, including quadtrees

Figure 5.2: Data Structures for DTM definition

Figure 5.2. Peucker (1980) suggests there are six basic data structures for the storage of a DTM. The most common used cartographically is contours (4), though the irregular points (3), allow for the true representation of data surveyed in the field.

The second format, regular triangular points, is less commonly created in surveying; more common is (3), the irregular triangular grid. This structure can be created directly from any three-dimensional survey with no conversion or interpretation of points. The triangulation of these points, often using the Delauney triangulation, allows an accurate surface model to be created based on the data available. This format is often combined with (5), three-dimensional strings, which are used as break points in the triangular data structure.

The most common way for three-dimensional information to be displayed on a two-dimensional map is that of (4), contours. These can be interpreted from the other forms of data illustrated in Figure 5.2. Contours are the most extrapolated data series and can bear little resemblance to the original data captured.

The methods explained above allow the landscape of a virtual model, as required by VHN, to be stored. The methods illustrated in Figure 5.2 do not easily allow for the modelling of multiple heights at the same location, or a spherical Earth. For many applications of spatial information it is this possibility, to display multiple information about a planar point, that will see the move from two-dimensional data sets to three-dimensional ones.

To model the entire Earth, Dutton (NCGIA 1990b) suggests the usage of a quaternary triangular mesh (QTM). At the lowest level, the surface of the Earth is mapped onto an octahedron consisting of eight equilateral triangles, the vertices of these triangles being at the poles and 90 degrees apart around the equator. The resolution of the mesh can be increased by subdividing each of the triangles into four new equilateral triangles. At level twenty the resolution, or triangle size, is approximately 1m on the Earth's surface (NCGIA 1990b).

The modelling of landscape features is something, as mentioned earlier, that the vector data structure as used in both CAD and VR packages allow. The restriction imposed by geographical packages, in particular GIS that have not had a CAD background, means this cannot be done. Even with those that have a CAD background, many of the GIS operations can only act on two-dimensional data. The concept of each x, y co-ordinate only being allowed one z co-ordinate is an idea that would seem very strange to a CAD or VR designer. The geographical world needs to encompass other computer areas around it that are also storing three-dimensional spatial data. One group that can store worlds that are three-dimensional and highly detailed is the computer games industry. Presently no technology used in the games industry is used in the geographical industry. Yet simulator games such as those for racing, flight and golf simulators regularly use very accurately surveyed terrain data in the games. The technology transfer has gone one way, but as yet has not come back.

Out of the formats mentioned in this section the one that lends itself best to the structure of the landscape information for the VHN project is the grid of regular rectangular points. This has two advantages in its structure that aid the subsequent display of the data by a virtual reality package. Firstly the landscape can be divided into tiles (in a similar way to how OS maps are) that are stitched together in the virtual reality software. Therefore, instead of having to process the entire landscape during display only those tiles that are visible from the current viewpoint are processed. Secondly, the tiles can contain varying representations that are displayed depending on their distance from the current viewpoint. This idea is discussed further in Section 5.5.

5.2.2 Raster Structures

Apart from the representation of a surface by a vector data model, it is also possible to represent surfaces by a raster approach, although it is less common to do this for large-scale data. The raster can be modelled as a grid of cells, called pixels, with height as an attribute. This representation can easily be created from the regular point grid illustrated in Figure 5.2. The alternative method to pixels is to create volumes of cells, called voxels, which have x, y and z co-ordinates, and therefore can display other information, such as land use, as an attribute.

The differing approaches of vector and raster to represent large-scale terrains can be influenced by the usage being made of the data. If only cross-sections of the landscape are required, then a raster approach will be quicker than having to calculate all the line intersections in the vector approach. The vector storage of a landscape, however, will generally be smaller than the same landscape in raster. However, with increasing processor speed and storage space on computers today these two restrictions are becoming less of a tie to which storage method is used. Generalisation techniques such as quadtrees and octrees, which can be applied to both raster and vector data, allow this storage overhead to be reduced. The amount relies upon the regularity of the data set or the resolution the user/application requires. Today the collection method and the application of the data more often determine the method used. This is why it is rare for large-scale data to be raster, as most raster collection devices, for example, aerial and satellite images, tend to be small-scale. This may change in the future, with large-scale laser scanning and high-resolution satellite imagery as discussed in Chapter 6.

The voxel approach can easily be extended to include features on the landscape at a level of detail, relative to the size of a voxel. It is difficult to model terrain features by

using of any of the data structures suggested in Figure 5.2 as these only allow for a single point at any one location. In order to include landscape features in a vector model then a new structure for the data is required. Voxel technology is applied more often in the medical imaging field (Bissell and Papageorgiou 1998) than for geographical data. Although in addition, some computer games, for example, TumbleBugs from Voxar Limited (Voxar 1998) use voxel graphics for their three-dimensional displays. This sees a move from the usage of voxels to model purely solid objects. In the medical and game applications, mentioned selected voxels are left transparent allowing the user to move through the object or environment. Increased display speeds are acquired, as the computer does not have to calculate the visibility of multiple polygons as in current games. The display is a mapping of a three-dimensional array of voxels to the two-dimensional screen. Currently however, the large data volumes associated with voxel models means that they are not suitable for web based applications where the information has to be transferred across the Internet.

5.2.3 Three-dimensional Primitives

With three-dimensional data, there is the possibility of storage by three-dimensional primitives reducing the vector information required for construction. For visualisation purposes, the best technique is to have a completely faceted three-dimensional model. Here, the best primitive to use is a triangular facet, as these can only lie in a single plane and therefore no data is lost about the internal definition of the facet. For display requirements, the number of facets that have to be computed and displayed can have a large influence on the system. Hence, if four points lie in the same plane it is more economical in data storage and subsequent display to store this as a rectangular facet rather than two triangular facets. At the current moment in time two-dimensional

mapping is represented by a series of one-dimensional objects, lines, and zero-dimensional objects, points. Therefore extrapolating this up a dimension, three-dimensional mapping should include the addition of two-dimensional objects, faces. Correctly in a three-dimensional environment, however, it can be argued that volumes should be stored, not faces. Therefore, solid objects should be represented by tetrahedra to ensure zero data loss. Due to current methods of data collection, this conversion to storing complete three-dimensional objects is not possible.

In terms of data storage of the objects, it is simpler to store regular geometric primitives for example cubes, spheres and cylinders, for which less information is required. It is then up to the viewing or manipulation software to handle the conversion between how the data is stored and displayed or used. For example, the minimal information to display a sphere as a primitive is the (X, Y, Z) co-ordinate of the centre, and a radius. Similarly, to store this as triangular facets requires at least six triangles for a simple shape. This value increases if more detail is required. This is the approach that VRML uses to handle file size, and is discussed further in Chapter 3.

For geographic information it is not often the case that the data exists in a format that can easily be converted into regular tetrahedra or other regular geometric primitives. Therefore, this format is of little use in the accurate representation of geographic entities. Where the format can be used in a project such as VHN is for the pictorial display of objects, such as street furniture or items inside the properties. The other application is in the representation of objects at a lower level of detail (Section 5.5).

5.2.4 Stereograms

An interesting way to store three-dimensional information, which has recently become popular, is the stereogram. In particular, single image random text and dot stereograms,

the same, except patterns of dots are used rather than characters (Figure 5.5). Therefore a much higher resolution image can be achieved and a greater range of depths.

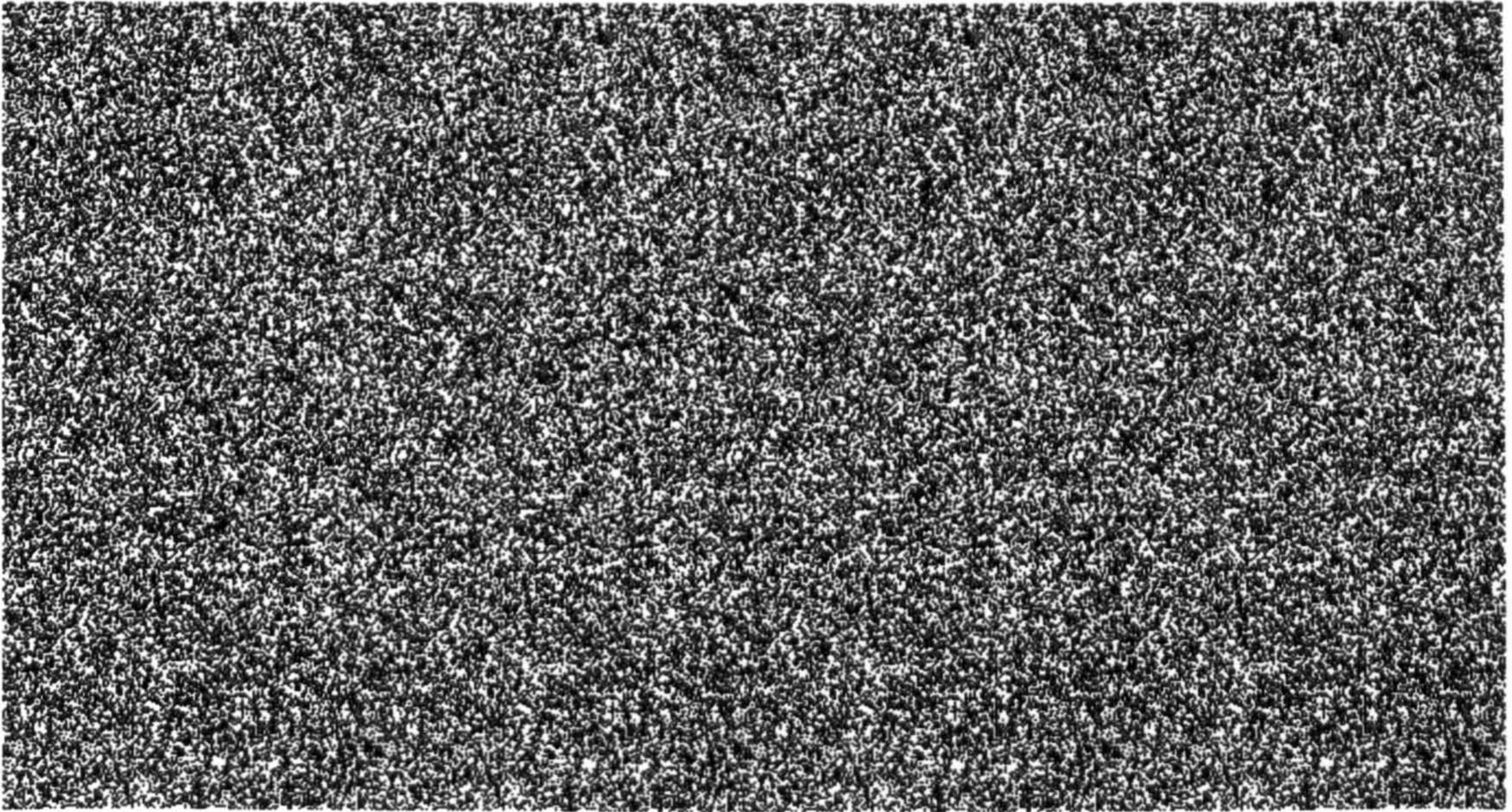


Figure 5.5: A Single Image Random Dot Stereogram

Figure 5.5. SIRDS can be used as a means of confusing the brain to see a three-dimensional image. These allow for far more detailed images than those created by SIRTIS explained above.

No research has been carried out to explore the ability of storing three-dimensional geographical information in SIRDS, although a glance through any of the Magic Eye books available shows that quite high detailed information can be encoded into the image. If the reason to store the three-dimensional information is for visualisation purposes only, then SIRDS represent a compact and easily displayable method of structuring the data.



Figure 5.6: Image used to generate previous SIRDS

Figure 5.6. Current SIRDS software uses the colours in the image to infer the depth of the objects in the SIRDS.

Figure 5.6 shows the original image used to create the earlier SIRDS. Current SIRDS software uses the colours in the image to infer the depth in the resulting SIRDS: here the darker the colour the higher the object appears. Through their inability to have true colour encoded into them, and lack of ability to attach further information to the image, SIRDS are of little use in a GIS application.

However, SIRDS can be a very effective tool for giving a feeling of three-dimensionality and depth to an image that can be displayed in basic graphics applications or even a printed image. The application of SIRDS to geographic data is dependent on a three-dimensional model being available from which to create the image. Therefore, for applications such as VHN where the data source is mainly photographic, three-dimensional models would have to initially be created using techniques such as Phoenix (Chapter 8). The three-dimensional effect of SIRDS is further enhanced when animated movies are produced using the technology. However, the effects on eyesight through straining the eye to see the three dimensions would need to be examined.

5.2.5 QuickTime VR

Although this has the title of VR software, it does not use a three-dimensional model as a basis. QuickTime VR, from Apple, is a raster approach to giving the appearance of a three-dimensional model. The data is captured by use of a series of pictures taken at various orientations from the same point of view. The two-dimensional raster images are then stitched together by the software to give the impression that the user has up to 360° of movement. The freedom of view depends on what images are taken, allowing the user to look left and right or up and down, an example image is given in Figure 5.7.



Figure 5.7: QuickTime VR model of Westminster Abbey

Figure 5.7. QuickTime VR stitches together a series of two-dimensional images taken from a single point to produce a 360° view. The example here is of Westminster Abbey and is taken from MS Encarta '99.

Hot-links, similar to the idea in HTML, can be added to allow the user to move around the scene or to find out further information about parts of the images. If done correctly, the user is given the impression of being in an interactive three-dimensional

environment. As only raster images are used, the data volume is kept small, compared to the required amount of three-dimensional vector data and information to produce the same scene.

For spatial information, little data is stored about the physical dimensions of the objects in the scene that could be used in a GIS application. QuickTime VR can be utilised as another means for displaying information to users where they are only concerned with visualisation. In this way, it offers a cheaper solution, in terms of data capture, and display hardware, than an equivalent VR model. For the purposes of the VHN project, little use can be made of the Quicktime VR approach, as sufficient images do not exist to allow for the creation of the three-dimensional scene. For the generation of visual representations of existing buildings, this approach provides a quick and cheap alternative to the more usual forms of three-dimensional surveying.

5.3 Four-dimensional Data Structures

When the application moves up into the fourth dimension, which will often be time, data storage is only available in VR style formats. In the geographic world, time is an attribute, not a dimension. Yet, some data sets, for example the weather, modulate in space with other dimensions, not just time, or with more than four dimensions. As of yet there is no way to model and store this apart from as (x, y, z) data and various attributes. Ideas have been proposed for the storage of three-dimensional data in geographical standards, but there is no move towards the storage of data with n dimensions. This is a concept which is often used in mathematics but which is very hard to portray visually and conceptually.

Current GIS data structures treat the fourth dimension as time, and therefore produce animated data sets where a three-dimensional view is presented for a particular time slice. In mathematics and geometry, the concept of the fourth dimension is not only time, it is another dimension similar to width, height, or depth. It is our current concept of space, which defines only three directions, that means we are unable to visualise the fourth dimension. One area where four-dimensional representations are used is in the definition of transformations between co-ordinate frameworks, by the usage of homogeneous co-ordinates. By defining locations in three-dimensional space by four-dimensional co-ordinates, it is possible to store accurately points that fall at infinity. The mapping from four-dimensional space to three-dimensional space is the same as that from three-dimensional down to two-dimensional space, as given in Equation 5.1.

$$\begin{aligned}(x \ y \ z) &\rightarrow \left(\frac{x}{z} \ \frac{y}{z} \right) \\ (x \ y \ z \ w) &\rightarrow \left(\frac{x}{w} \ \frac{y}{w} \ \frac{z}{w} \right)\end{aligned}$$

Equation 5.1

Barnard (1983) states that homogeneous co-ordinates were initially an analytical tool for projective geometry, but have since been applied to computer graphics and industrial automation.

In the novel *Flatland*, Abbot (1884) suggests an interesting way of allowing the fourth dimension to be structured and visualised. By examining how a two-dimensional world would see the third dimension, it is possible to extrapolate how we may see the fourth dimension. If a three-dimensional object were passed through two-dimensional space, a plane, then an observer confined to two-dimensional space would see a series of two-dimensional objects. These would possibly vary in shape and position for the duration

of time that the three-dimensional object passed through the plane. In this way, to the two-dimensional observer, time becomes the third dimension. Therefore, a four-dimensional object will cause a three-dimensional shadow which varies over time when passed through our space. Figure 5.8 uses a stereopair of images to illustrate an example of this phenomena.

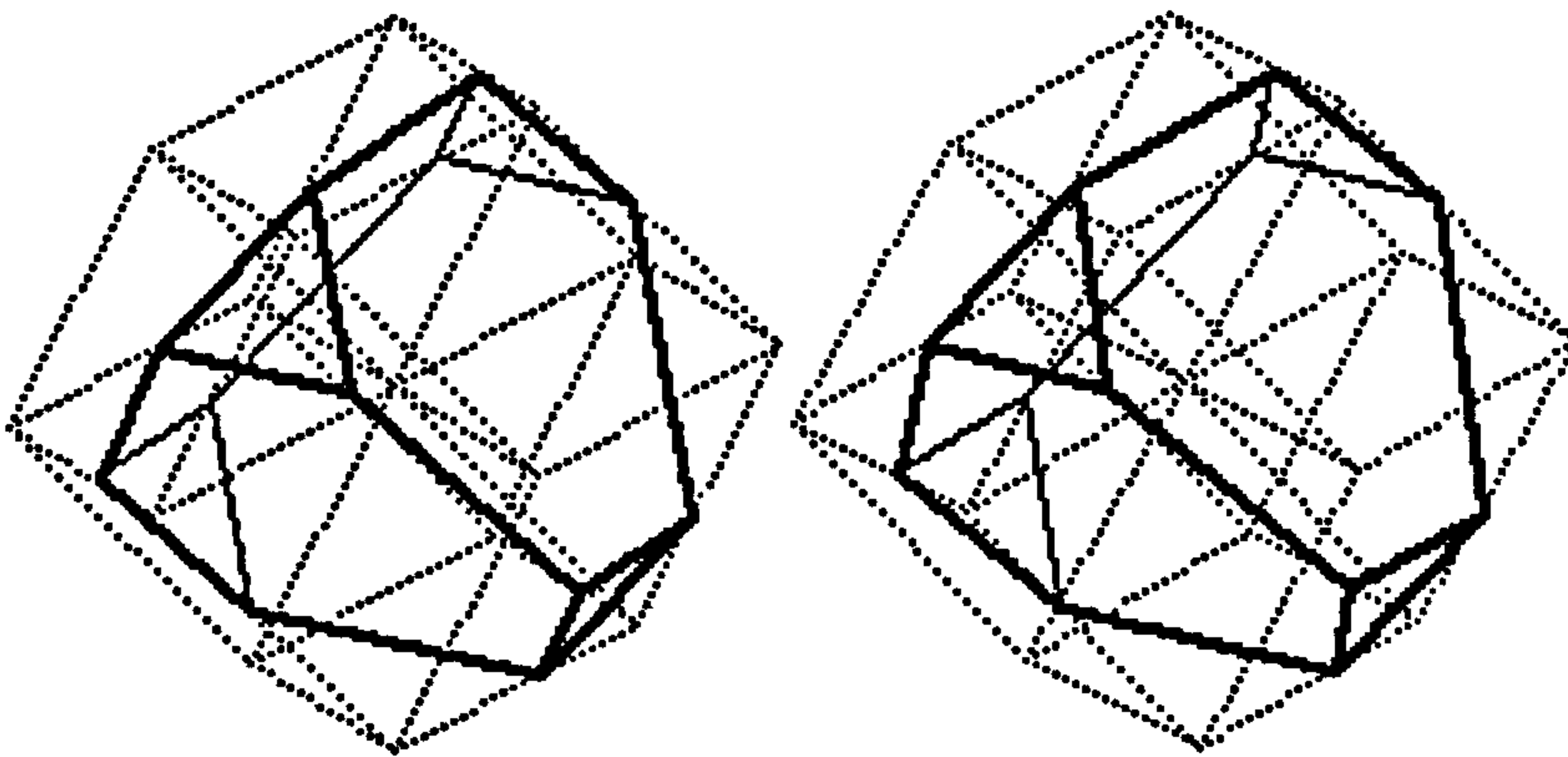


Figure 5.8: A stereopair of a four-dimensional hypercube

Figure 5.8. The solid is an intersection of a four-dimensional cube with three-dimensional space, a hypercube slice. It penetrates space orthogonally in the direction of its hyperspace diagonal. The dotted lines are the shadow of the four-dimensional cube projected onto three-dimensional space. The stereopair is best viewed by focusing behind the page (Scheller 1997).

These regular n^{th} dimensional shapes, or polytropes, can be further represented by other methods. For example, it is possible to unfold a three-dimensional cube into the six two-dimensional squares that construct it. Similarly, a four-dimensional hypercube can be unfolded into the eight cubes it is constructed from (Hausmann 1994). Therefore, the four-dimensional object can be stored as a template of three-dimensional objects that construct it.

Current general spatial information and software make little use of the fourth dimension except for specialist applications, such as climate monitoring. In the spatial arena the fourth dimension often refers to a fourth variable rather than to time, and therefore a multi-dimensional GIS does not specifically have to be one that is able to handle time.

Those that can handle time often do not extrapolate between the discrete data values entered.

5.4 Object-Oriented and Hierarchical Structures

As mentioned in Chapter 4, the computing world is moving towards an object-oriented approach to computing. This not only affects the programs but also the data that passes between those applications. If we begin to consider data as objects, or collections of objects, then the ability to pass these between differing applications is eased. Not only is the data passed to the application, but also all the methods that belong to that data. In a perfect object-oriented environment any data could be displayed, printed, integrated, or edited inside any application. At this stage, the boundary line drawn between data and application becomes less distinct than it is today. As data is included in an application, the available functions you can perform on that data is increased by the methods included in the data. By allowing data objects to inherit and contain other data objects, a more hierarchical structure applies to data than at present. Prototype objects can be built, and instances of these then used in the data set, thus reducing some of the repetition in current data sets.

These ideas are being developed for spatial data, such as the VRML language that is an object-oriented based data description language. The idea and structure of VRML is covered in more depth in Section 3.4. Objects in the VRML world are represented by prototype objects with differing attributes. These prototype objects can be simple geometric shapes, for example spheres or cubes, or they can be complex collections of objects such as a table, house, or robot. Similarly, these objects can then have methods attached that are fired in response to events. For example, the robot may respond to a time event that moves it around, and the house could contain a door that opens in

response to a mouse click event. In this way knowledge is built into the model which is data dependent not application dependent. It is the data, the door, that knows what to do when it is clicked on. The application does not know that the object called 'door' should open when clicked. This allows 'dumber' applications to be built but data sets that are more complex have to be constructed to account for this.

By building object-oriented data sets the objects, rather than entire data sets, can be passed between applications. This would allow separate components to utilise the same object of data, without having to be able to access it from the data set. In this way, it becomes possible to embed these data objects inside other containers and objects.

Embedded files or objects are one way of enabling data sets to become increasingly intelligent faster. If in one model you create a door object, as described above, then it only exists in that one file. In moving to an object environment, then this door can just be created as a single object, which may be stored as a file. Then this object can be embedded into many other objects. Therefore, you can have a house object, which contains a door, and a factory object that contains the same door. Objects now begin to consist of more of these basic object types. For the house, we may embed the objects door, wall, window, floor, and roof, all of which contain information for what should happen if a user touches them, walks into them, looks at them, etc.

An extension of this idea, building complex objects from simple objects, can also be taken a stage further for symmetrical objects. It is possible to store just a section of the object and then details of the rotational or planar properties of the symmetry that are used to create the complete object. Although this saves on file storage for complex objects, the display time is increased, as the computer has to generate the model each time it is displayed. For many applications of large-scale geographic information there

is not much duplication in the data set to allow for the creation of, for example one door, that can be used for all the houses in a town: this is especially the case with historic data as with VHN. Where the ideas of splitting large models into smaller, hierarchical, objects can aid in this case however is in the automatic generalisation of the model.

5.5 Generalisation and Levels of Detail

The main problem with creating a three-dimensional representation of the world is the volume of added data, compared to the two-dimensional map that is currently used. In three dimensions, often mappers feel obliged to include extra detail such as street furniture, vehicles, and textures. All this adds to the load on the processor to display the environment many times a second.

The concepts of generalisation in two-dimensions is well formulated with various approaches available, for example the Douglas-Peucker algorithm. These allow for the generalisation of linear features by a formal method of removing points along the feature that are not required for it to retain its shape. Though these exist, little software automatically employs these techniques when displaying data on to the screen. Current systems are optimised for drawing information on the display and therefore it is often faster to draw all the detail than to calculate what information to remove. This is because often the linear feature not only represents a drawn line, but also connections, and these must be kept during generalisation.

The main form of generalisation for objects in three-dimensional space is level-of-detail (LOD). Therefore, at a distance a house may be a simple extrusion of the floor plan of the building, or made from geometric primitives. As the user moves closer, more detail

in the floor plan and roof appears on the building. Moving even closer, outlines of doors and windows begin to appear. When the user reaches the object, these windows and doors become more detailed and intelligent; including handles that open and windows that are transparent (Figure 5.9).



Figure 5.9: Example of levels of detail of an object

Figure 5.9. An object can be given various levels of detail so that the object gets increasingly detailed as the user viewpoint moves towards the object. This aids the display rate of the entire model and goes towards keeping the number of facets displayed at a constant rate.

The idea of LOD's in a three-dimensional object is analogous to the concept of generalisation in two-dimensional mapping. However, in three-dimensions the generalisation is caused by moving the viewing position relative to the object. Little software exists now that can produce generalisation 'on-the-fly' when viewing a three-dimensional scene of an object. One piece of software that can generate generalised models is LODestar from the Technical University of Vienna (Schmalstieg 1997). This

allows for the generalisation of VRML files based on the amount of data loss required, allowing a series of automatically generated models to be created from the original highly detailed model (Figure 5.10).



Figure 5.10: VRML model of a ship being generalised by LODestar

Figure 5.10. The original ship is generalised by reducing the overall polygon count of the object as the specified viewing distance increases from left to right (Schmalstieg 1997).

As can be seen from the figure, the change between the first three levels of detail is hardly noticeable to the human eye. Although, as can be seen from Table 5.2, the polygon count has decreased from 4698 to 2990 (a reduction of 1708 polygons, or ~36.5 %).

Table 5.2: The results for a model generalised by LODestar

LOD	Polygons	Statistic	Range
0	4698	(quad=0 tri=4698 line=0)	1962.509644
1	4142	(quad=0 tri=4142 line=0)	2664.944092
2	3686	(quad=0 tri=3686 line=0)	4070.779297
3	2990	(quad=0 tri=2981 line=9)	6882.473633
4	1486	(quad=0 tri=1478 line=8)	12505.423828
5	444	(quad=0 tri=435 line=9)	23706.371094
6	112	(quad=0 tri=108 line=4)	46122.359375
7	24	(quad=0 tri=24 line=0)	90932.835938
8	0	(quad=0 tri=0 line=0)	infinity

Table 5.2. Full details of the models produced by LODestar when generalising the model shown in Figure 5.10. The LOD's shown in Figure 5.10 are 0, 3, and 6 from left to right (Schmalstieg 1997).

What Figure 5.10 also illustrates are the effects of applying the LODestar algorithm to the whole model as opposed to sub-objects. For example it would be possible to make

level six look more like the original ship if the sails and the body of the ship were separate objects. In this case, it is likely that the sails would generalise down to simple planes while the body of the ship would approximate to a rectangular prism. This approach of influencing the generalisation results in an object that retains more of its spatial characteristics.

This solution towards generating LOD's aids in the visualisation of the model and not in its storage size, as for each object multiple representations have to be held in the data file. For the idea of LOD's to be improved, it is necessary to develop techniques and algorithms to allow fast and efficient generation in the viewing software, rather than the authoring software. In this way, the data file would only have to contain the most detailed model of each object, and information on what percentage to reduce it by for different distances.

One idea suggested by this research for the rate of increment in the polygons, is to return to the suggestions made at the beginning of this chapter, those of following nature. This is, that over a set distance the amount of polygons should follow a Fibonacci sequence (or some multiple of) for example,

$$8 \rightarrow 13 \rightarrow 21 \rightarrow 34 \rightarrow 55 \rightarrow 89$$

Therefore, although the distance between the stages will remain constant, the difference in the number of polygons will increase. It would be necessary to test this theory and see if it gave users the impression of smooth change between the LOD's.

As stated earlier in this chapter, the application of these techniques to a townscape model allows the user to get a greater feel for the virtual world. Many applications simply cull objects at a specific distance from the position of the user. In a flat

environment this technique works fine as often objects in the foreground obscure those behind. In the VHN application, as can be seen from Figure 6.2, the townscape contains a hill. If this information is culled, then when the user is down in the valley near the river, information is lost. Instead, by reducing the data across the model through the approaches of levels of detail and the culling of smaller objects, then the impression of the hill remains.

5.6 Application of a VR data structure to GIS data

Fairbairn and Parsley (1997) describe the application of a VR data structure, VRML, to four example sets of GIS data, consisting of

- small-scale map (approx. 1:250,000) of the Tyneside conurbation, in Northeast England
- medium-scale map (approx. 1:35,000) of the city of Newcastle upon Tyne
- large-scale map (approx. 1:6,000) of the central campus of the University of Newcastle upon Tyne
- interior building survey obtained using photogrammetric plotting of room photographs

These data sets were used to create essentially VR representations of two-dimensional maps for the small- and medium-scale data, and three-dimensional worlds for the large-scale data and building survey. In order to simplify them for conversion, all text, and some extra detail was removed from the maps. The conversion involved the creation of three-dimensional faces for every ground line, thus 'raising' the buildings by the correct value. For the large-scale map, the building heights were determined using stereo photogrammetry from aerial photography.

To reflect the changes in scale between the maps, levels of detail were set up between the small- and medium-scale maps. Thus, as the user moves closer to the large-scale map it triggers the replacement display of the medium-scale map. This allows, in effect, for a generalisation operation to be called at an appropriate stage in the navigation. However, as stated in Section 5.5, this generalisation is decided in the data set rather than the software. This move between levels of detail is sudden and can be unnerving if the change occurs when the user is too close to the object. The smooth transition of objects currently relies on complex multiple levels of detail being generated inside the data structure, in turn increasing the file size.

Two other methods were used for transferring control of the display from one world to a more appropriate one. As with HTML, 'hyperlinks' can be embedded in a VRML scene, attached to objects, and link to other objects, text, images, sound files or complete worlds. The use of hyperlinks was also applied in the example, in order to attach attributes (notably the names of buildings) onto the objects. Dummy links were set up; allowing text to be rendered on screen as the cursor passes over it. Similarly, transparent objects were placed around areas, with the links and text attached to them.

A final method of linkage, supported by VRML 2.0 browsers and of considerable potential, is the insertion of sensor nodes. These can automatically call different and more detailed representations if the navigator moves close to a 'hot spot' with a link to other scenes. Such sensor nodes are not presently capable of being automatically created, requiring hand coding in the VRML language.

Fairbairn and Parsley (1997) discuss the problems that arose while converting from two to three dimensions when moving from a linear based data set to a face based data set. In their example, polylines (chains of lines) in the initial mapping had to be converted to

individual lines for the conversion process. Furthermore, the faces created needed to be contiguous, in order to obtain separate areas (for example of parkland and buildings on the map) whereas current linear based mapping is discreet. The conversion process can be time-consuming, but allows individual areas to be altered, for example; raising parts slightly above the background in order to make them visible. The concept of priority in layering, as in a GIS, does not exist in VR, as it is designed to handle three-dimensional, rather than two-dimensional data. Without the introduction of a z-value to each face, both the background and the face can interact, giving unpredictable graphic results, dependent on the viewpoint.

5.7 Summary

Some of the problems discussed in this chapter result from storing essentially two-dimensional data in a three-dimensional structure. Therefore software interpreting the data set expects it to have basic three-dimensional characteristics; i.e. height. The concept of layering data in a three-dimensional structure could still have benefits, not only in pure visual display as above, but also in deciding the priority with which to draw objects at speed. Currently, few software applications accurately decide on-the-fly how to reduce detail in an object while the user moves, often resorting to flipping to a wire-frame display for movement or random culling of geometry. By the introduction of layering, the software would be given some idea of the importance of objects in the world, and be able to more readily decide on what to cull during movement.

As shown in the chapter, many aspects of data structure, for example either increasing the speed of display or decreasing the size of the storage, affect different properties of the object. The structure of the data should be controlled by its use, not the software

that uses it. Until open computing allows software to read multiple formats easily, the majority of data will not be stored in the most optimal format for its use.

An application such as VHN has to trade off all these factors against each other when deciding on a data structure to use. As has been shown in this chapter the structure can be influenced not only by the data type, but also on requirements for file size and the usage. In some cases, the structure of the data is decided at the time of capture. Chapter 6 goes on to discuss the current methods of data capture.

6. Three-dimensional Data Capture

GIS was born with two dimensions due to the history of the GIS and its origins in paper mapping. Early systems were simply a way of digitally displaying the equivalent of a paper map. Unfortunately this limitation has stayed until recently, with GIS developers concentrating on analysis and remaining with the two-dimensional view of the world. This is reflected in the data capture field where, even with increased digital technology, such as the electronic theodolite and the global positioning system (GPS), position is often recorded and processed separately to height. Position and height are frequently recorded to different accuracy; with usually only the object footprint on the landscape recorded.

As stated earlier, in many ways the data is the most important part of the GIS. For applications, such as VHN, to be able to move to three dimensions it is necessary for the data to move to three dimensions at the same time, if not before. This does not necessarily mean that existing data is obsolete, as it is possible to combine existing data sets to produce three-dimensional ones (Section 6.1). What it does imply is that data collection and storage should now be carried out in three dimensions, rather than being reduced to two.

The OS have moved to a three-dimensional network with the new GPS network that is replacing the original system of triangulation pillars for position and benchmarks for height. Basic GPS techniques are good at fixing position but poorer at giving height, often resulting in only position being recorded. The tools now exist for the display of three-dimensional data, with future developments around the corner. Unfortunately, GIS data models cannot gain from this due to their missing third dimension at the data

capture stage. Sinning-Meister *et al.* (1996), suggest that for three-dimensional data sets

“...the degree of detail and accuracy of the model must be balanced with the amount of data to be manipulated (possibly on-line and desirably in real-time).”

Three-dimensional data models utilise both two-dimensional and three-dimensional data. Many people still require a two-dimensional plan view to orientate successfully in three-dimensional space (even when considering the real world and not a simulated one). People often become lost if immersed at ground level in three-dimensional data. Charitos and Rutherford (1996) quote Smyth *et al.* (1994) who examine how two-dimensional maps are used

“You tend to hold it [the two-dimensional map] in one way and if you are travelling along, you may have to rotate your map in order to try and find the correct direction to go, because the task of doing the rotation in your head is difficult.”

Bourdakis (1996) suggests that the ability to jump or fly above a three-dimensional model can aid in navigation, effectively allowing the user a more two-dimensional view of the data. In the long term, the two-dimensional map should just become an orthographic projection of the three-dimensional data model, rather than a separate data set.

Chapter 5 explained how three-dimensional data could be stored; this chapter examines approaches to how that data can be captured. This chapter offers a review of the various techniques currently used for capturing the three-dimensional shape of an object or landscape. Following on from this Chapter 7 will describe techniques using a single

image to produce a three-dimensional model of an object and, therefore, this concept is not discussed in this chapter.

The data capture technique used in the field often depends on the accessibility of the area, the equipment available, the accuracy of measurement required, the data format required and the time available for data capture. Therefore, it is not possible to state one method that should be used for data capture, but it is necessary to have knowledge of those available in order to be able to fit the technique to the task. Some of the methods illustrated in this chapter have the ability to be used for either buildings or landscapes, and some capture both simultaneously.

This chapter aims to discuss current methods that are used for the capture of three-dimensional geographic information and their applicability to projects such as VHN. These techniques do little to aid in the development of townscapes of historic Newcastle, but do enable the creation of an accurate and detailed view of present day Newcastle to illustrate the changes over time. The chapter also introduces some new capture techniques currently being developed that will aid in the transition to a three-dimensional data model. The theme of data capture is continued into the remaining chapters of this research (Chapters 7, 8, and 9) where the problem of interpreting three-dimensional information from a single oblique image of an object is discussed.

6.1 Extension of existing data

The majority of data sets currently used at a large-scale are buildings in two dimensions, and landscapes in two-and-a-half dimensions. This is for two reasons; firstly, this is how most of the data has been supplied, and secondly the problems the data has been

produced for are two-dimensional. The main three-dimensional application of data is line-of-sight calculations.

To enable data sets and problems to grow to the third dimension, rather than supplying new data sets, existing data can be extended and combined to form the three-dimensional world. The information to create a three-dimensional data set exists in the data available today, what is lacking are the tools to carry out automatic conversion, the GIS to use it, and the problems to apply it to. In many ways it has become a circular argument, with the data not being captured as the software is not there for it, and the software not being developed as the data is not there to use it. However, as mentioned in Section 2.1, GIS that can handle three-dimensional data arrived on the market in the latter part of 1998.

The ability to convert, automatically, a data set depends upon the structure of the two-dimensional data, and the additional three-dimensional information available. Currently OS large-scale data sets are supplied as two-dimensional 'spaghetti', or link and node data. In order to use this in a three-dimensional environment, objects have to be built from the line data and then be extruded to give an indication of their three-dimensional structure. Proposals for a new Landline data format, originally referred to as OS96, are face and object based. This would allow for easier automatic conversion from current two-dimensional mapping to a three-dimensional map. Unfortunately, four years later it does not appear that this format will be used and little reference is now made to it by OS.

As stated above, existing data can be given a three-dimensional appearance by extruding a two-dimensional feature along its z-axis. Although the features will all remain on a planar surface, the visual effect can be satisfactory for many applications.

The appearance can be further enhanced by including the automatic generation of roof outlines. These can be created from the skeleton of the floor plan of the building, as pictured in Figure 6.1. The skeleton is created by buffering inwards the outline of an object until only linear features remain. By raising the lines, as shown in Figure 6.1, the impression of a roof is created.

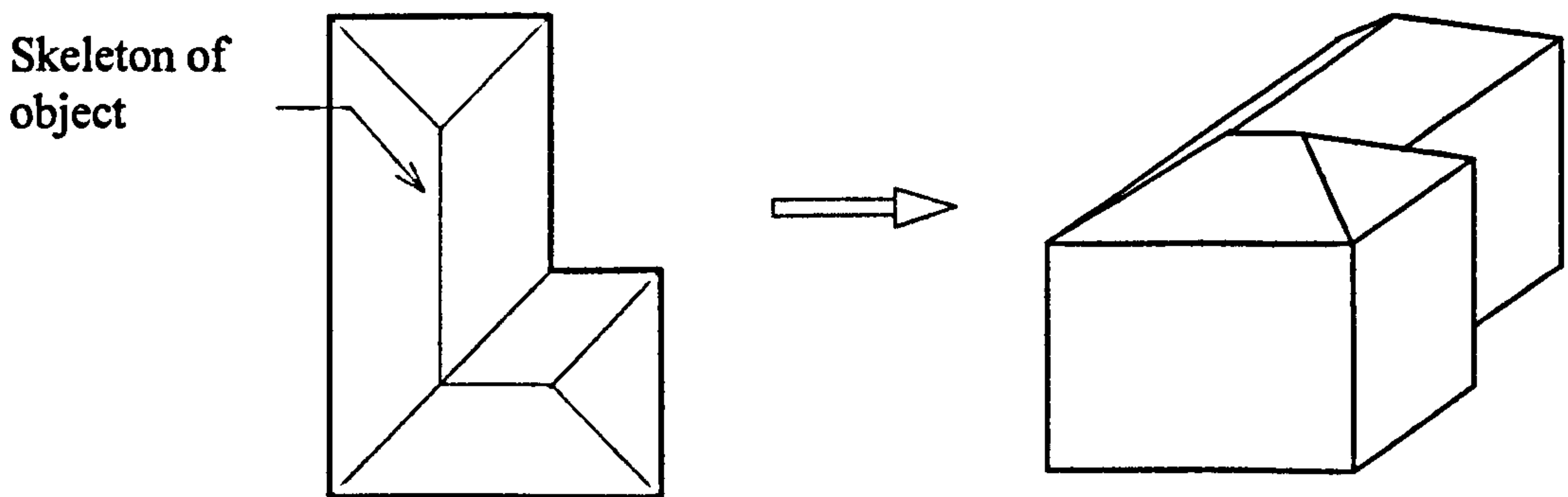


Figure 6.1: Usage of an object's skeleton to create a roofline

Figure 6.1. The skeleton of a two-dimensional building can be used to create a roofline, giving a more realistic appearance to the three-dimensional object.

Although this does not accurately portray the roofline of a building, if used globally across a data set, the visual impact of a building with a roof compared to a plain block is greater. It is important to realise that this addition is no different to the current forms of symbolism used in two-dimensional data. For a more accurate model, other data sets can be combined with existing spatial data sets to aid in the construction of a three-dimensional model, for example Cities Revealed (Section 2.2), includes information on the height of buildings and roads.

Figure 6.2 shows an example of these principles applied to the historic data available for the Quayside area of Newcastle upon Tyne used by the VHN project. Here, the building outlines from the historic map have been extended and then placed on a historic landscape. Although this model is very simple, it gives a greater sense of realism than the equivalent paper map. This allows the viewer to gain an easier

understanding of spatial factors, for example the location of roads, which can be seen to run along the bottom and up the side of the main mound.

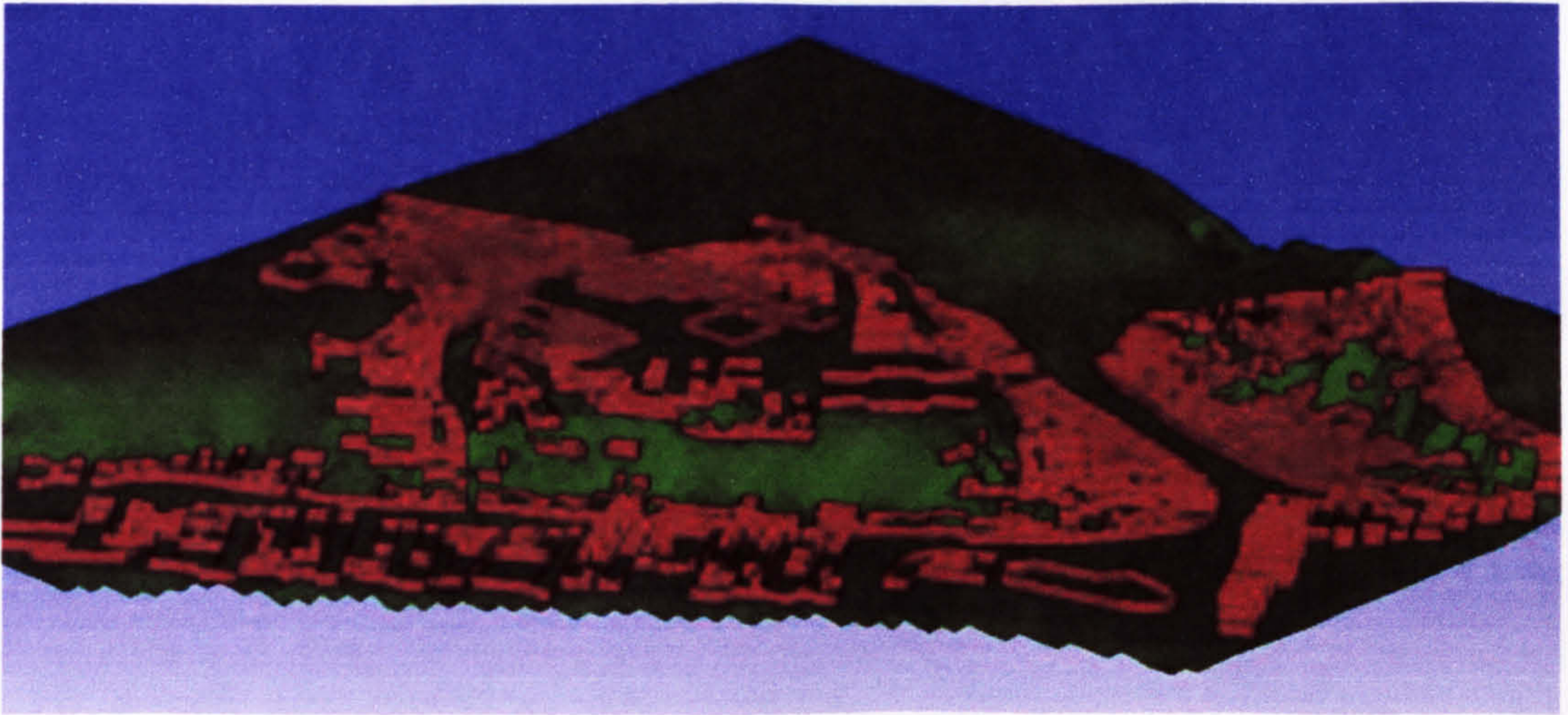


Figure 6.2: Use of extending existing data to produce a 3D townscape.

Figure 6.2. Existing historic data of the Quayside is used to create a simple three-dimensional townscape by extending the building outlines and placing them on a historic landscape.

Even without a high level of detail on either the buildings or the landscape, more of a feel of the geography of the area is given than is immediately gained from a two-dimensional map. Further enhancements can be made by using the new high resolution aerial and satellite imagery, discussed in Section 6.5, draped over a terrain model to give a greater sense of reality. To many users, a three-dimensional visualisation has to appear more realistic than the equivalent two-dimensional map although it may only convey the same amount of information.

6.2 Theodolite survey

Where data does not already exist, a new survey should consider the three-dimensional structure of an object as the final product, rather than the current acceptance of a two-dimensional plan view. The most common ways (in 2000) to survey the three-dimensional structure of an object are; using a total station (a combined theodolite and

electronic distance measurement (EDM) set-up) or stereo photogrammetry (discussed later in Section 6.5).

The collection of data for a three-dimensional model requires that the survey method should be altered in order to aid the post-processing of the data before storage. Current surveying looks at capturing linear information, for three-dimensional information it is better to pre-consider the creation of faces and objects in the survey of data. The survey process involves a network of control stations created around the building or landscape to be surveyed, which are traversed to give a high positional accuracy. Angles and distances can then be measured to various points of interest on the object or landscape from these control stations. Each measurement is taken from at least two control stations, though further measurements will add redundancy to the data set and aid in the final solution. Further measurements are taken in order to build up the detailed view of the object. For a landscape, it is necessary to survey the change of shape of the landscape as well as the points of detail on the landscape. Many terrain data sets include either the detail with little information on the height of the land, or vice-versa, therefore not allowing the creation of a true three-dimensional model.

With the survey of a building, it is often only possible to place a prism, the target at which the theodolite is pointed, at ground level. For full building capture two options exist, laser-ranging systems (Section 6.3), or flat adhesive targets applied to points of detail on the building. Often when using targets for a complete survey of a building, for applications such as VHN or the testing of the Phoenix algorithms (Chapter 9), too many points of detail are required to apply targets to all of them. To avoid having to do this, only angle measurements are recorded to fix the points of interest above the ground

level. The information is then analysed using a least squares approach to calculate the three-dimensional co-ordinates of the points.

6.3 Laser ranging systems

An improvement to the use of a total station for surveying objects, is to move to a system where a prism or target is not required to gain a distance measurement. Laser ranging systems work on the principle that the surface of an object reflects the laser light back towards a receiver, which measures the time (or phase difference) between transmission and reception in order to calculate depth. The technique used is similar to a traditional survey, explained in Section 6.2, except that there is no requirement for the placing of reflective targets on the object to measure distance.

Difficulties arise when targeting the laser point on the same position from multiple locations. New systems allow for the remote control of the survey equipment, enabling the operator to be closer to the façade of the building they are surveying. However, for some locations it is still impossible to get to the required control point. Other improvements to this technique can be made by using two theodolites and the operators targeting on the laser dot supplied by each of the systems. These systems still require the targeting of the individual points of detail required for the model and are therefore not practical for applications such as VHN where an entire townscape is required.

Systems exist for use on smaller objects; these scan the entire object to produce a point-cloud that can then be surfaced to produce a solid object. For example, the ModelMaker system (3D Scanners 1998) works by scanning a laser line across the object that a camera then picks up, as depicted in Figure 6.3. The position and orientation of the camera are known, allowing for the three-dimensional shape of the

contour to be calculated by the Surfa module. These contours are then combined to form the complete model. The technique used in the capture of the information for an object in these systems means the final data sample can be very large.

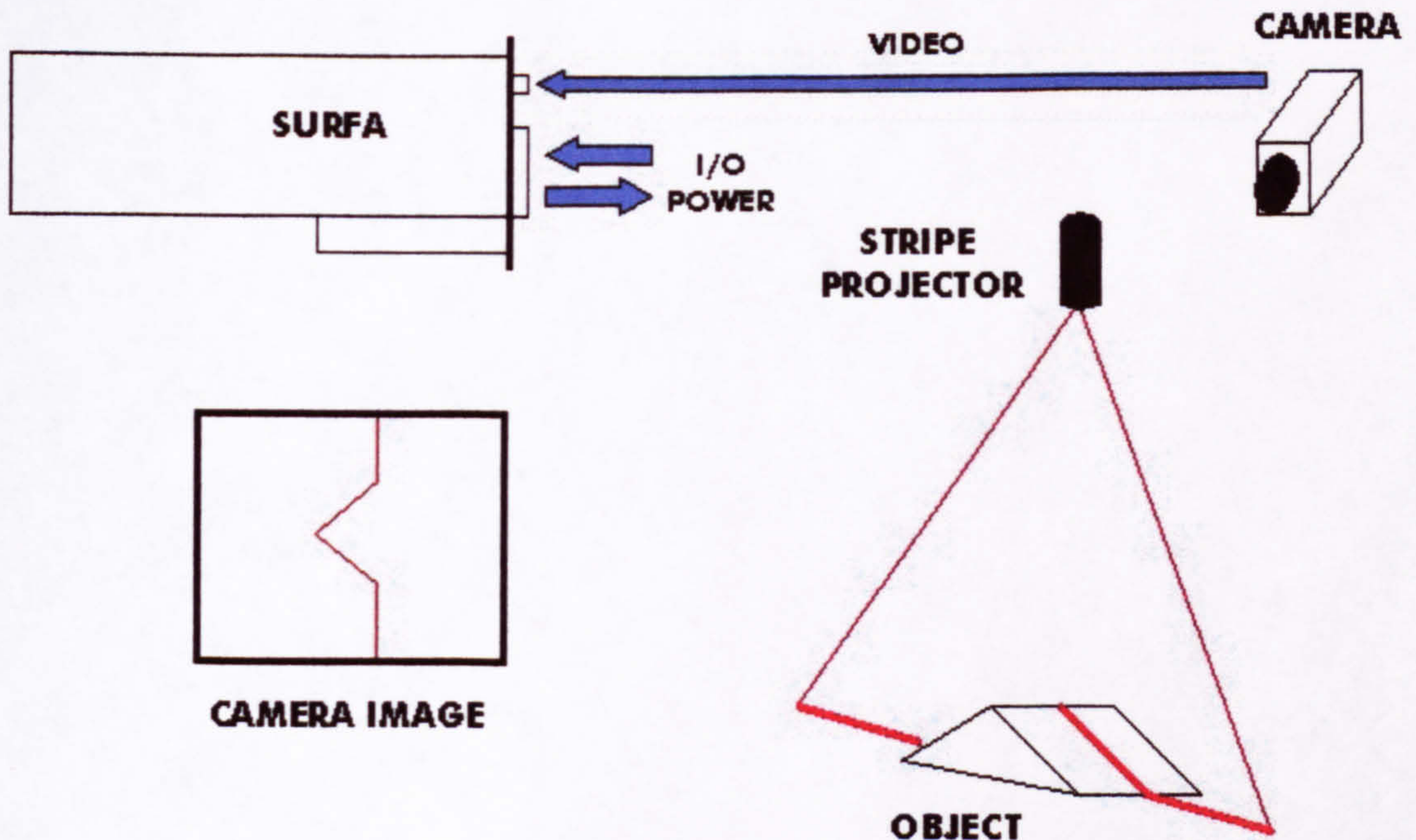


Figure 6.3: The ModelMaker system

Figure 6.3. The ModelMaker system works by scanning a line of laser light across the object. A camera mounted with the laser records the contours produced. These are then combined by graphics hardware to form the final model (3D Scanners 1998).

The use of a handheld scanner restricts the size of object that can successfully be scanned and surfaced to produce a three-dimensional model. The use of airborne LIDAR scanning means that it is possible to carry out a similar process to that above, when scanning terrain from an aircraft (Hoss 1996, Flood and Gutelius 1997). This captures both the landscape and the objects on the landscape using a point scanning method. A dot is scanned across the landscape and, in a similar way to the distometers and theodolites mentioned earlier, the distance is calculated from the returned phase difference, or travel time. The process builds up a point-cloud of three-dimensional data based on the position of the plane and the distance to the object on the ground. This is illustrated for modern Newcastle upon Tyne in Figure 6.4.

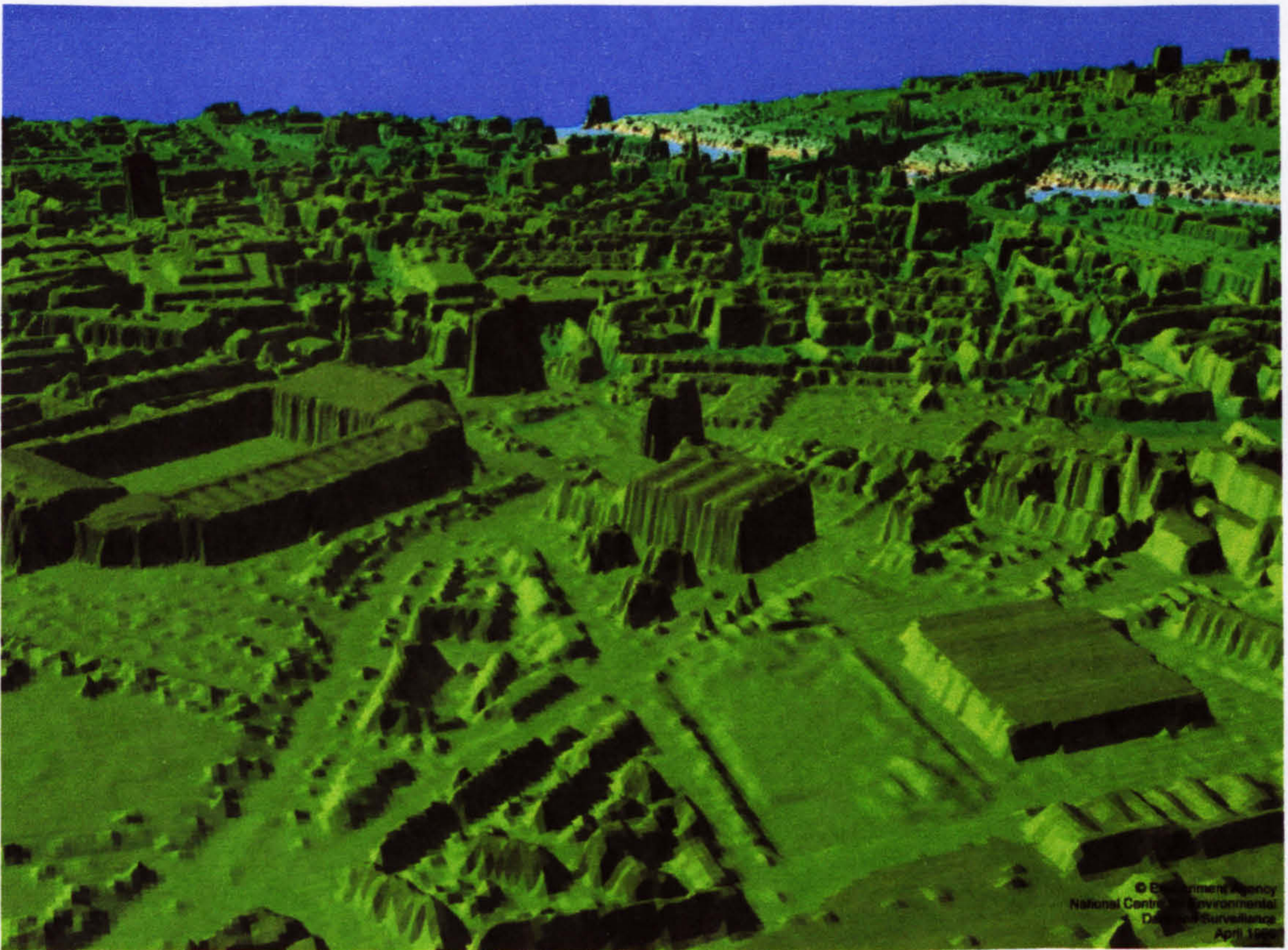


Figure 6.4: LIDAR data for Newcastle upon Tyne

Figure 6.4. Shows an example of LIDAR data for Newcastle upon Tyne looking down from the centre of Newcastle towards the Quayside. The point cloud has been given shadows to aid in the visual interpretation.

While this technique produces a model that can be used visually, the data does not contain any information about what it represents. For this data to be of use in the creation of three-dimensional models, the point-cloud can be combined with large-scale digital mapping to delineate the areas of the model. This system has the advantage that not only is the landscape captured but also the objects upon it, for example the height and shape of St. James' park can be seen on the left in Figure 6.4. Airborne scanning allows for the fast generation of townscapes when combined with existing large-scale mapping to aid the interpretation.

6.4 Satellite Positioning

GPS surveying uses the position of a group of satellites to calculate the distance, or time, to a receiver on the ground. The accuracy of GPS depends on the data capture technique and the equipment used. The accuracy can vary from 100m down to the millimetre level in plan. Improvements to the stand-alone position can be made using differential GPS to eliminate some of the errors in the calculation (Table 6.1).

Table 6.1: Comparison of errors for stand-alone and differential GPS

Error Source	Stand-alone (m)	Differential (m)
Satellite clock	15.0	0.1
Ephemeris	40.0	1.0
Orbit	5.0	0.13
Ionosphere	12.0	1.0
Troposphere	3.0	0.5
Multipath	2.0	2.8
Receiver noise	0.5	0.7
Total root sum squared	44.8	3.3

Table 6.1. Differential GPS techniques, where corrections are broadcast from a base station to other receivers, allow the total error to be reduced from 44.8m down to 3.3m (Quest 1996).

Differential GPS works by broadcasting the errors calculated at a point with known co-ordinates (the base station) to other locations and the receiver. As can be seen from the table, the first five sources of error are reduced when using differential GPS compared to stand-alone. However, two of the error sources, receiver noise and multipath, increase for the differential technique. These errors occur as a factor of the receiver and its position. Therefore, for differential GPS, where two receivers are involved, these values will increase. Multipath error is caused by the signal from the satellite reflecting off other objects before reaching the receiver, causing a time delay and thus a difference in location to signals that reach the receiver directly. The multipath error detailed above in Table 6.1 will also increase in built up areas due to the increased problem of line-of-sight to the receivers from the satellites

Another technique used for the enhancement of position is kinematic GPS; this can reduce errors to the centimetre level. Kinematic GPS requires the position of a base station and rover receiver to be initialised before the rover is allowed to roam. Initialisation involves both receivers fixing on the same satellites for their readings. The systems must then keep track of the satellites so that the lock is not lost, and only a few cycle slips occur, otherwise the set up has to be reinitialised. Detailed three-dimensional models of the terrain can quickly be built up using kinematic GPS, and these can be interpolated to give data for inaccessible areas. This type of system is already in use in the tracking of emergency vehicles, stolen cars, and security vehicles. Kinematic GPS cannot only be used to capture full landscapes, but also has applications in tracking the three-dimensional position of a moving object.

The use of GPS for the capture of buildings is not applicable due to the large multipath errors that occur in built up areas. As with theodolite survey (Section 6.2) GPS still requires the user to visit each point of interest for which a data point is required. GPS survey is applicable for an application like VHN in the fast collection of landscape data when compared to theodolite survey.

6.5 Stereoscopic methods

Stereoscopy is a technique for measuring range by triangulation to selected locations in a scene, imaged from two camera positions. This technique is used to generate three-dimensional models from aerial photography of landscapes, and for close-range photography of buildings. Close-range photogrammetry describes the technique used when the object to be measured is less than 100 metres in size and the cameras are positioned close to it (Cooper and Robson 1996). This makes close-range photogrammetry an ideal candidate for the capture of the three-dimensional shape of

buildings. The advantage of photographic techniques is that they are not required to touch the object, and a permanent record of the data used for the measurements exists. A stereopair can be produced with a single camera from two locations, as in aerial stereoscopy, or by the use of a stereometric camera. A stereometric camera consists of two metric cameras mounted at both ends of a bar that has a precisely measured length, generally 40 or 120cm (Doneus 1996). The two cameras are used as projectors to display the images on the base plate below them.

There are three stages to set up a pair of photographs for measuring, these are referred to as inner, relative and absolute orientation. Inner orientation involves setting the projectors to have the same properties as the cameras at time of exposure. Relative orientation constrains the relative position and rotation of the projectors to each other. Finally, absolute orientation positions the projected images in their correct three-dimensional location from the usage of at least three control points surveyed in the photographs. The control points are locations easily identifiable on the ground and the image for which an accurate position is known. With most stereoscopic instruments, the user positions a floating dot onto the same point in each image, allowing the three-dimensional co-ordinates to be read off the instrument. Again, as with the theodolite survey approach, inaccuracies can occur from failure to position the separate points on the same location.

Cooper and Robson (1996) state that the usage of digital cameras for close-range photogrammetry has allowed the transfer of machine vision algorithms and concepts into photogrammetric processes. This allows features to be automatically identified, matched and transformed into three-dimensional features in object space. New technology that supports this automatic generation of objects from stereo images is the

range of applications from Tricorder. The Tricorder Professional is a hand-held system that allows photogrammetric scanning of complex small to medium size objects (Tricorder 1998). This permits the creation of three-dimensional models at the location, rather than having to record and later process the data. In this way, errors or omissions in the model can be caught at an earlier stage.

Recent improvements to the quality of satellite based images has led to an increase in the data available for large-scale capture. Capes (1998, page 25) states that

“... laws governing the availability of VHR [very high resolution] imagery been relaxed to allow almost unrestricted distribution of material with a 1m ground resolution.”

The information from this new generation of Earth observation (EO) satellites to be suitable for mapping up to 1:5000. Although not suitable for producing accurate large-scale drawings at a 1:1250 scale, these VHR images add a wealth of detail to enhance existing 1:1250 data. Figure 6.5 shows an example of the imagery now obtainable from the IKONOS-1 satellite, which has a ground resolution of one metre. Satellite imagery of this resolution allows a greater continuum of data than from an aerial survey operation, as it is periodically collected.



Figure 6.5: Imagery of San Francisco, California from the IKONOS-1 satellite

Figure 6.5. New satellites such as the IKONOS-1 can produce a ground resolution accuracy of 1m, this allows mapping at scales of 1:5000 (Space Imaging 1998).

Table 6.2 illustrates the current availability of commercial satellite images. Correction can be applied in various levels depending on the usage of the data. Capes (1998, page 25), describes the corrections detailed in Table 6.2

“Low-level correction comprises systematic removal of image distortions caused by the curvature of the Earth and the acquisition angle of the satellite. High-level correction comprises registering imagery to a map reference system, removal of topographic distortion to create ortho-images and the creation of digital terrain maps.”

Table 6.2: Summary of EO data characteristics

Name	Data type	Ground res. (m)	Coverage (km)	Max scale	Product prices
Space Imaging's IKONOS-1	high res. optical	1 (pan) 4 (XS)	11 x 11	1:10000 to 1:5000	~\$20-30 per sq. km (low-level correction) up to ~\$80-90 per sq. km (high-level correction)
QuickBird	high res. optical	0.82 (pan) 3.28 (XS)	3 x 3 up to 20 x 120	1:10000 to 1:5000	~\$20-30 per sq. km (low-level correction) up to ~\$80-90 per sq. km (high-level correction)
OrbView-3	high res. optical	1 and 2	8 x 8 up to 84 x 40	1:10000 to 1:5000	To be decided but estimated to be ~3% less than the cost of aerial photography and cheaper than its competitors' data
KVR 1000	high res. optical	2	40 x 40	1:10000	\$4000 per image
ERS	radar	12 (max)	100 x 100	1:100000	\$1800 per image
Radarsat	radar	9 (max)	50 x 50 up to 100 x 500	varies	\$3250 per image

Table 6.2. Earth observation satellites allow mapping to be produced at a larger scale than today's satellites (Capes 1998).

The capture of stereo images allows for the creation of a digital elevation model (DEM) from the imagery. This includes the height of the landscape and the objects, such as buildings, on that landscape. Capes (1998, page 24), states that "DEM's derived from one metre resolution satellites should allow vertical accuracy of between two metres and three metres". This accuracy is outside the usual accuracy for large-scale mapping, with OS 1:1250 maps having an absolute accuracy of plus or minus half a metre (Havercroft 1997). However, the resolution of 1m is superior to the DEM's available from the OS (50m for OS Profile data). Further advantages come from the cost of obtaining the data

compared to a traditional land or aerial survey. Capes (1998), suggests a reduction in cost from £3000 for a traditional survey of a 1km square, down to £75 using EO data.

For the fast and direct construction of two-dimensional mapping it is better to have high level correction applied to the images. However, for the creation of three-dimensional information through further stereo work, no correction need be applied. The accuracy of townscapes produced this way will increase with the resolution of satellite images, increasing to around half a metre in 2004, through the launching of new IKONOS satellites (Loeb 2000).

6.6 Light methods

Light techniques can help in the capture of the three-dimensional shape of a smooth plain object where it would be difficult to discern shape using photogrammetry alone due to the lack of feature points. These techniques involve altering the light that falls on an object, either through altering the angle and intensity or by applying filters, so that a texture is given to the surface.

The structured light technique calculates the surface shapes of an object from the distortions of patterns of light, for example, grids, stripes and elliptical patterns, projected onto the object. This method is very simple to set up, though has a low spatial resolution, as the patterns become sparser with distance. This makes the technique's application to buildings minimal, as its use would be very difficult to capture an entire building. However, for close range work on architectural detail or historic artefacts good resolution can be obtained.

The Moiré fringe method is an extension of the structured light method explained above. The main difference is in the usage of a reference grating as well as the

projection grating used for the structured light approach. The essence of the method is that a grating is projected onto an object, and an image is formed in the plane of some reference grating, shown below in Figure 6.6.

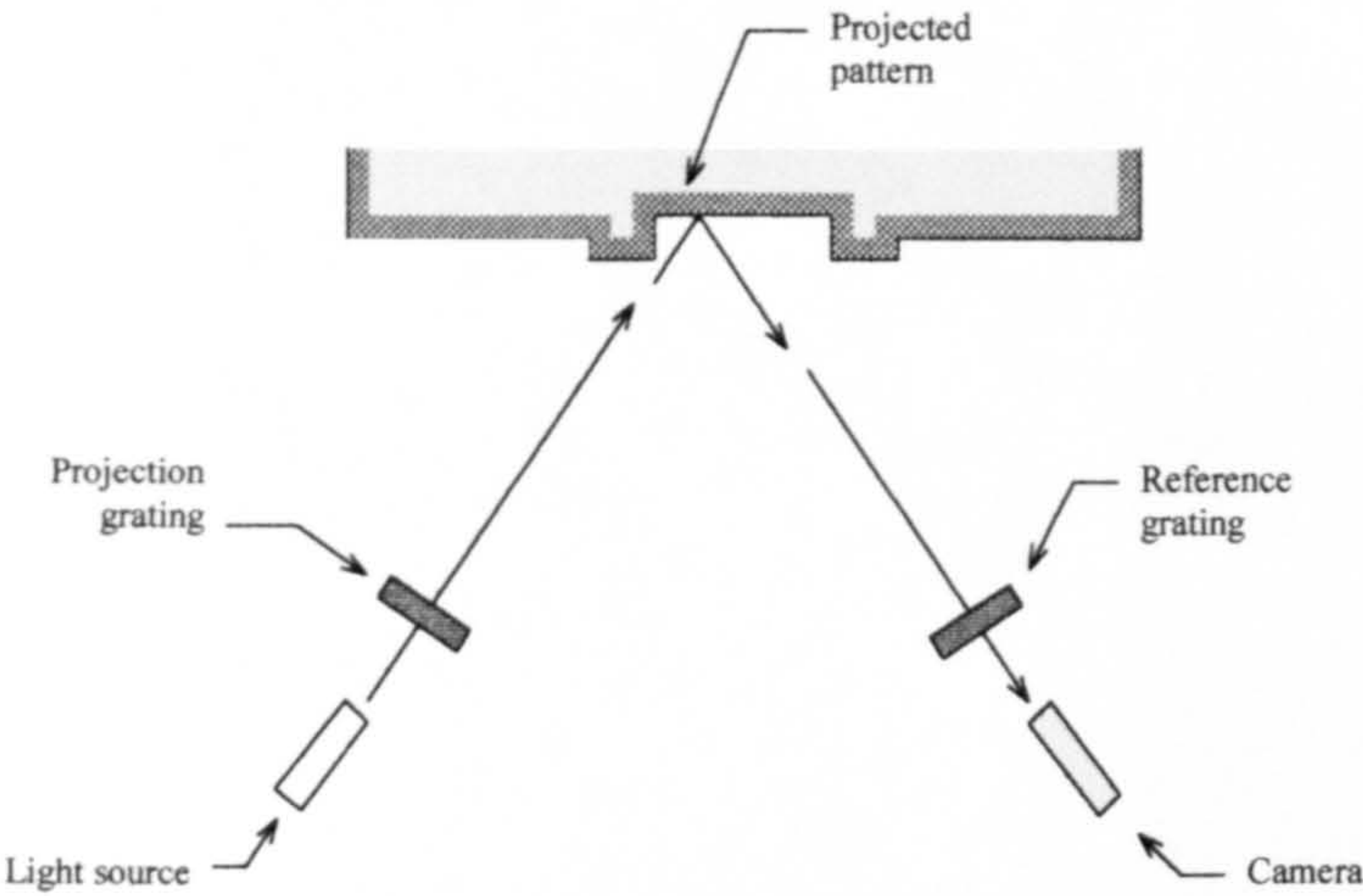


Figure 6.6: A moiré projection system

Figure 6.6. A grating is projected onto an object and an image is formed in the plane of a reference grating. The image then interferes with the reference grating to form moiré fringe contour patterns (Marshall 1997).

The image interferes with the reference grating to form moiré fringe contour patterns that appear as dark and light stripes (Figure 6.7). Analysis of the patterns produced gives an accurate description of change in depth, and hence reveals the objects shape.

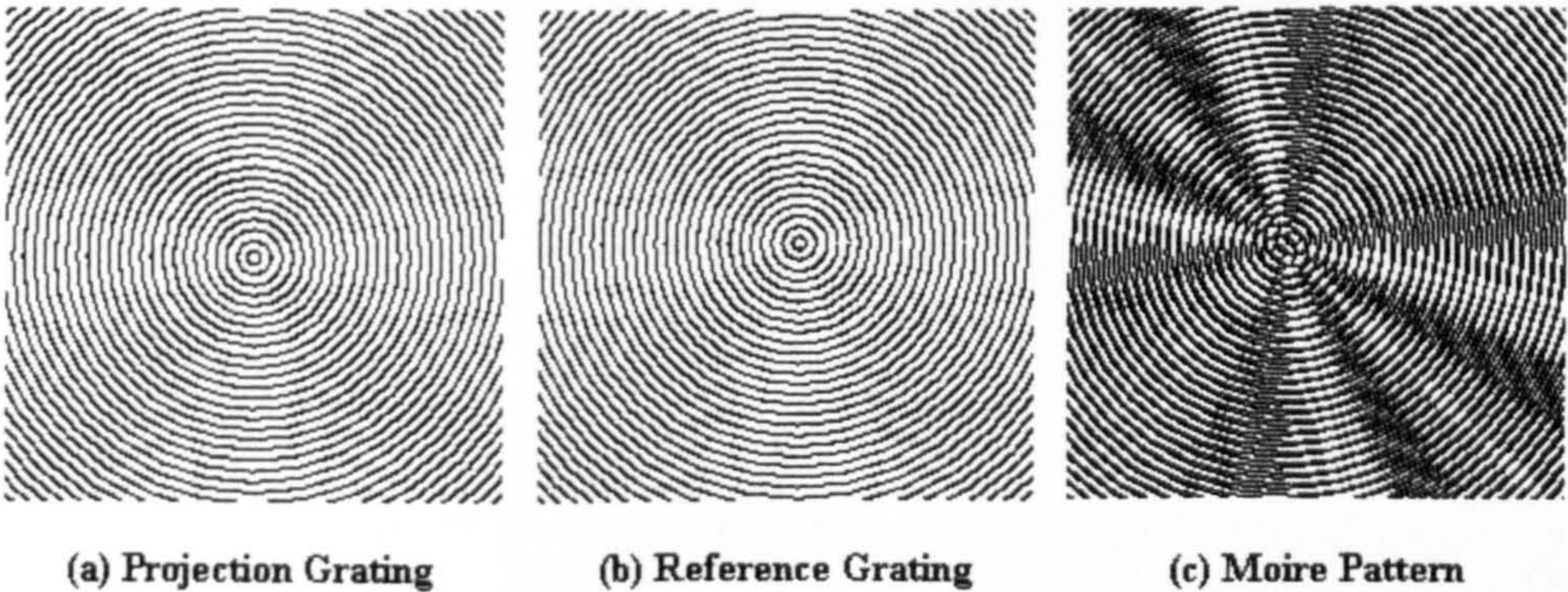


Figure 6.7: Moiré fringe patterns

Figure 6.7. When the projection grating (a) and the reference grating (b) interfere, moiré fringe contour patterns appear as dark and light stripes (c). Analysis of these patterns gives an accurate description of the change in depth, and hence reveals the shape of an object (Marshall 1997).

A single moiré pattern produced in this way, does not help to deduce whether adjacent contours are higher or lower in depth. Moving one of the gratings and taking multiple images can solve this. The effects of the reference grating can be simulated by software, requiring only one grating to be used in the field. This also aids in the calculation of the extent of depth, as the position of the simulated grating can be altered digitally to produce varying moiré patterns from which relative depth can be inferred.

A commercial system to use this technology for the creation of three-dimensional models of objects is the InSpeck-3D (InSpeck 1998). This system uses the phase shifted projection moiré method, where the projected fringe pattern is moved during capture of the object. The three-dimensional shape is then calculated by examining the phase value at each point in the image. Figure 6.8 shows an example fully textured model created by the InSpeck-3D system. The system allows for fast data acquisition and processing of 0.3 seconds for a 640x480 point image.



Figure 6.8: Model creation by the InSpeck-3D system

Figure 6.8. The InSpeck-3D system uses moiré fringe patterns to generate fully textured three-dimensional models of objects (InSpeck 1998).

The InSpeck technique produces higher accuracy than the structured light method, but is more computationally intensive in determining the shape of an object. As before, to

cover an entire building with this technique would further increase the complexity of the survey.

Methods based on shape from shading employ photogrammetric stereo techniques to produce depth measurements. A single camera is used to take multiple images of an object in different lighting conditions. Studying the changes in brightness over a surface and employing constraints in the orientation of surfaces enables the calculation of depth information.

Shape from shading only works well on an object with a uniform texture and reflectance. For usage in an uncontrolled environment, this technique does not excel; consider the effects of the sun on lighting. In addition, this method makes it difficult to infer the absolute depth of the object, and generally only gives surface orientation. This allows the shape of an object to be extracted but not its size and position. The use of structured light techniques in the VHN project are therefore best kept to small objects, for example historic artefacts that are placed inside the buildings, or for areas of fine detail on the outside.

6.7 Multiple oblique imagery

A growing technique that allows the fast generation of three-dimensional models of objects is the use of multiple oblique imagery. This builds on the principles of stereoscopic methods, allowing models to be created from less accurate imagery due to the increased redundancy in the process. For its application to VHN data, one of the limiting factors is that many of the buildings have either one image, or multiple images from the same scenic viewpoint.

The creation of an object from multiple oblique imagery is a two-stage process. Firstly, at least a 1:1 relationship between points on the images has to be obtained, most commonly done by user selection. The second stage is to use these points to calculate the relative position of the cameras, and hence allow the distance from the cameras to the points on the image to be calculated. In general, the solution for the second stage can be obtained using an iterative approach if photogrammetric co-ordinates exist for at least five points. However, Longuet-Higgins (1981) shows that, if eight points are known, a model can be obtained from the solution of a set of simultaneous equations.

The use of multiple images is now one of the most common forms of non-stereo reconstruction, especially with low cost software such as PhotoModeler available for the PC market. This software requires the photographs to be well spaced around the image so the corresponding detail points form independent solutions to the model. The use of PhotoModeler for the survey of a building is further discussed in Chapter 9, where the technique is compared to other, more traditional, methods.

Use of multiple oblique imagery has the advantage of fast data capture, as little control is required in the positioning of the photographs. In addition, as images exist for the complete model, textures can be extracted from these for use in a three-dimensional model. As mentioned at the beginning of the section, the restriction of this method for VHN with the use of historic data is the lack of images from enough viewpoints to produce tight enough models. This restriction leads to the investigation and creation of techniques for use with single oblique images, further discussed and tested in Chapters 7, 8, and 9.

6.7.1 Shape from video imagery

An extension to the idea of multiple oblique imagery is the use of video imagery of an object to capture its shape. The technique is similar in principle to that of oblique imagery. The user indicates on the first frame of the video the photographic coordinates of the points. Software then tracks these points through successive frames, treating each frame as an individual photograph of the object, therefore, creating a series of still images containing corresponding detail points from which to create the model. At any stage the user can then also add or delete points that are to be followed by the software, speeding up the data capture and analysis compared to multiple oblique still imagery. The use of video also allows the processing of more images, giving a tighter control framework, though these images are at a lower resolution than photographic film.

6.8 Summary

This chapter shows that the method used for data capture can vary due to many parameters; for example, accuracy, time, cost, and the size and shape of the object being surveyed. Traditionally, buildings and landscapes have been captured using, theodolite intersection, photogrammetry (stereo or orthographic, close range or aerial) and GPS. In other disciplines, such as machine vision and in the area of Geomatics, other techniques are used to capture the three-dimensional shape of an object. Many of these are more suited towards the capture of smaller objects than an entire building or landscape. However, for large-scale applications such as VHN, the researcher suggests that there is no reason why these techniques cannot be used for detail on buildings. Where possible capture of new data sets should be surveyed in three dimensions. If the science of spatial information is going to move into the third dimension, the data it uses has to begin to move first.

For the VHN application developed as part of this research, the techniques introduced in this chapter only allow the generation of the present day model. This only helps to illustrate the changes that have occurred to the landscape and the buildings on it. Little use can be made of many of these data capture techniques when applied to the historic model, where the landscape and many of the buildings no longer exist or insufficient imagery is available for processing. The next three chapters build on the ideas introduced in this chapter to develop a method for capturing the three-dimensional shape of an object using only a single oblique image.

7. Single Image Methods

Following on from the photogrammetric methods described in Chapter 6, the approach discussed here is to use a single oblique image (which may be cropped) to produce a model where no control points exist. Here the term image is used to refer to paintings and drawings, as well as photographs. This is because information on many historic buildings is not available as a photograph. Current photogrammetric approaches for single images mainly use an orthographic, or rectified, photograph where the image plane is made parallel to a vertical plane of the model.

Single image techniques were originally required in this research due to the historic nature of the prototype townscape produced. For many of the buildings in the area studied, only a single image exists. Where multiple images do exist, these are often found to be from the same scenic viewpoint of the building, because the images were not meant for modelling purposes but for artistic illustration.

This chapter initially examines the geometry of the perspective image and what information it contains. Then explores the relationship of vanishing points to the image, object, and the camera. Building on this foundation, graphical techniques for extracting information from a single image can be produced. Finally, further techniques based on research into machine vision and artificial intelligence (AI) are discussed.

7.1 Perspective Geometry

For any image in two dimensions, infinite possibilities of objects in the three-dimensional world exist. The methods discussed in this chapter examine approaches to restricting the number of possible objects to just one. These work by imposing restrictions on the model with the human brain aiding the interpretation. Humans

interpret three-dimensionality from a two-dimensional image using visual clues and a priori knowledge of objects, for example, how walls intersect on buildings. However, the brain can be confused as in the case of optical illusions, as demonstrated in Figure 7.1 and Figure 7.2.

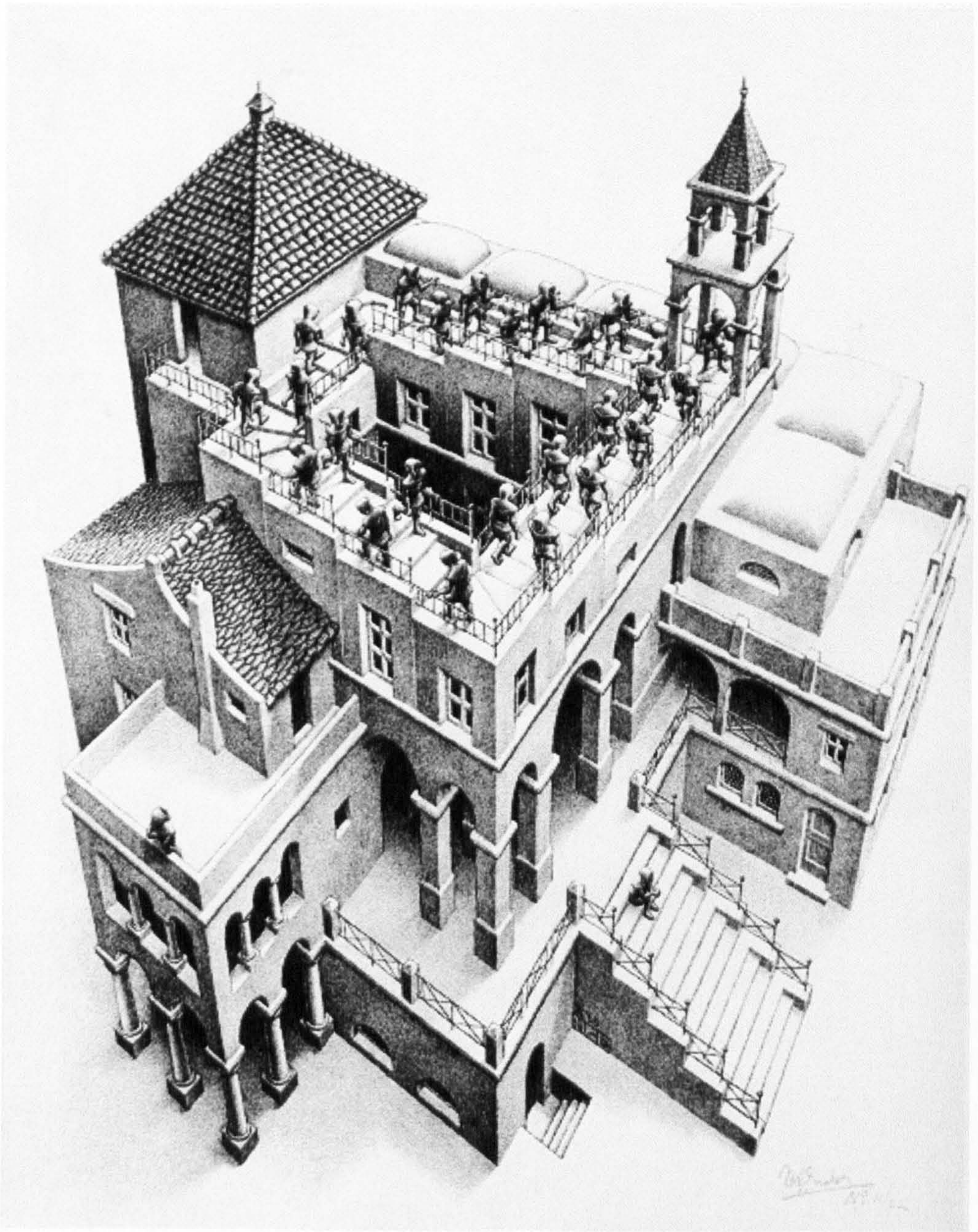


Figure 7.1: 'Ascending and Descending'

Figure 7.1. This illusion by Escher is caused by the artists' usage of false perspective clues and other visual stimuli, for example, the height of the guards, shading on the surfaces and distorted angles.



Figure 7.2: Hayward's 'Undecidable Monument'

Figure 7.2. Hayward's 'Undecidable Monument' shows the full power of false textures in the position of the boats and the shadows on the surfaces.

In 'Ascending and Descending' by Escher the figures can climb, or descend, the stairs on top of the building without getting higher or lower. Similarly in Hayward's 'undecidable' monument, a different three-dimensional model is suggested depending on whether the viewer restricts the viewpoint to the top or the bottom of the model.

These illusions are created by the use of false perspective, textural, and lighting clues, which suggest a different three-dimensional model to that possible. The other factor in this is that the brain will always see the most probable object based on the viewer's experience.

7.2 Mathematics of projection

In the interpretation of an image of an object, it is necessary to recreate the size, shape, and depth of the object (Barnard 1983). The size of the object is defined by Euclidean metrics (i.e. length, area, and volume) and is independent of the choice of co-ordinate system. Shape is defined as the spatial arrangement of the contours and surfaces of which an object is composed. The simplest geometrical contours are straight lines and planes. Whereas size and shape are measured relative to the object, depth is relative to the observer. Barnard (1983) continues to state that these geometric properties of an object are affected by perspective in two ways. Descriptive properties of an object, such as the collinearity of three points or the coincidence of three lines are invariant, whereas metric properties, such as length and orientation, are not. This information can be used to aid the reconstruction of possible three-dimensional shapes from a two-dimensional image.

The mathematics of projection is well formulated, and standard formulae have been produced for converting a three-dimensional object to a two-dimensional image (these are detailed further in Chapter 8). How an object appears in an image depends on its projection. In art, two basic projections are used, parallel, and central. However, many other forms of projection exist that convert from three-dimensions to two, with different properties, for examples consider the variety of map projections.

For a parallel projective transform (or orthographic), the rays from the object are usually perpendicular to the image plane (Figure 7.3). This is a good approximation for an image over a small area and away from the image's extremities.

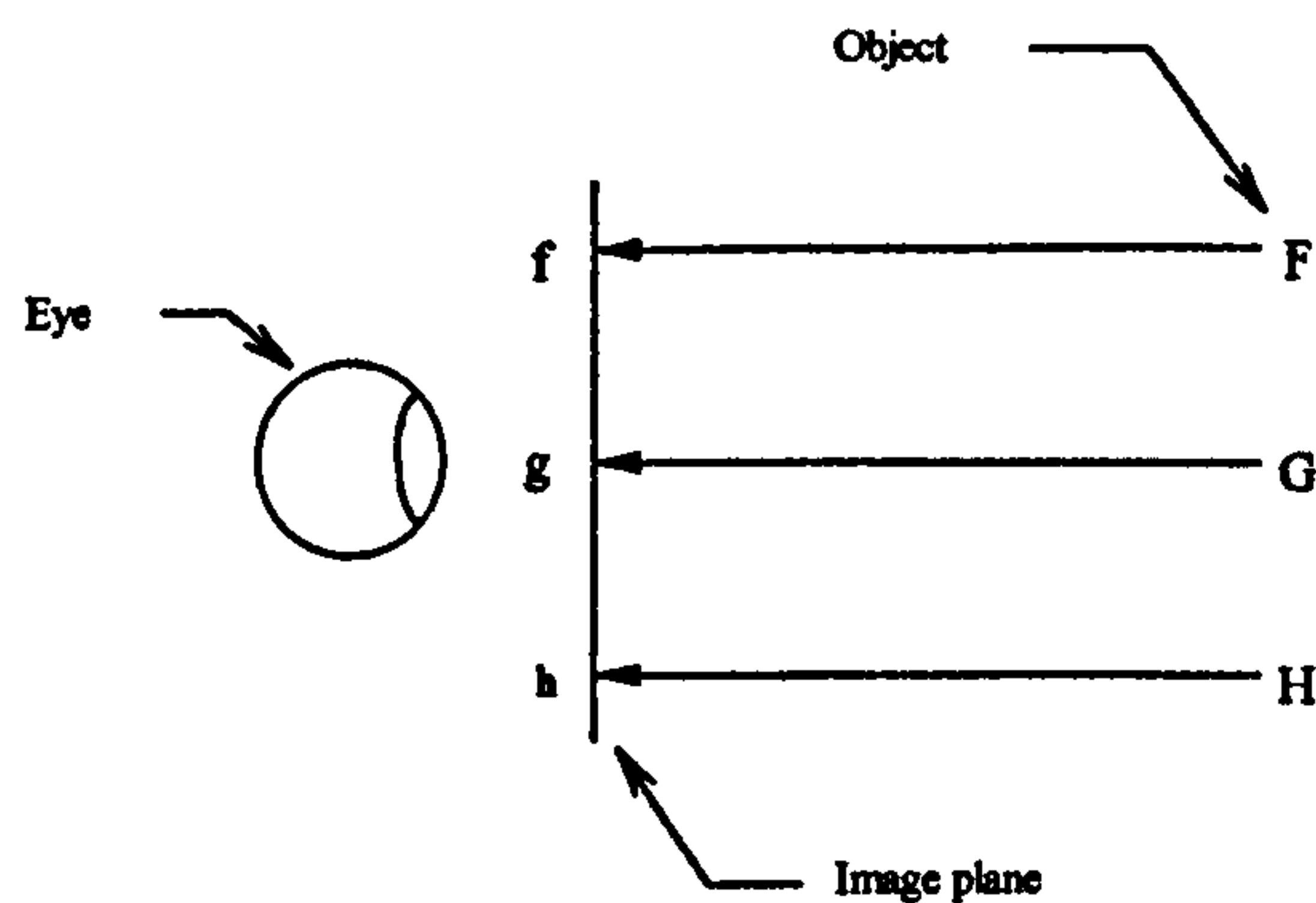


Figure 7.3: Parallel projective transform

Figure 7.3. The parallel projective transform, or orthographic, view of an object is created by constructing lines parallel to the image plane from the points of interest on the object.

The central projective transform (or perspective) forms a better model of an image, as shown in Figure 7.4. Here, the rays from the object to the image plane pass through a common point, or lens, called the focal point. This results in the image being reversed on the back of the eye or camera.

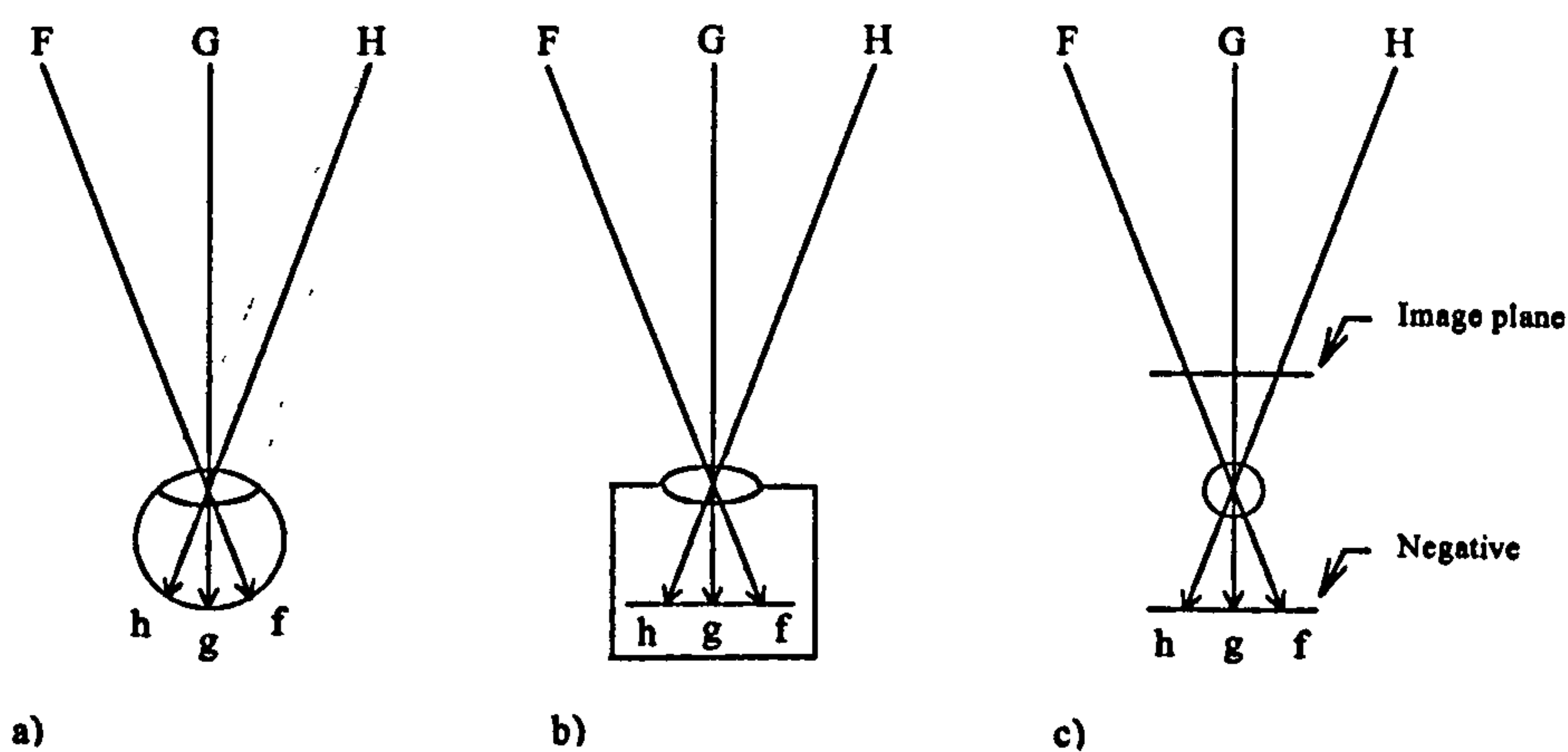


Figure 7.4: Geometry of the eye (a), the camera (b) and perspective (c)

Figure 7.4. The lens at the focal point of the transform affects the projective transform of the human eye and the camera. Lines are drawn from the point of interest through the focal point and then onto the image plane. This is best modelled using the central projective transform, or perspective.

When an image is produced from a negative, the image produced moves to the other side of the focal point. The geometry of perspective assumes a perfect lens at the focal point. In this research, there has been no attempt to include the modelling of lens distortions in the calculation of information. However, in order to make the technique more accurate it would be necessary to account for these distortions.

7.3 Vanishing Points

On a perspective image, it is possible to locate vanishing points, where parallel lines on the object will be seen to converge. The position of these points is a direct result of the position of the camera relative to the object at the time of exposure. Thus, if they are found the camera position can be recreated from them using a graphical or mathematical method.

Figure 7.5 shows a simple example object, and the location of various vanishing points of the object. The main vanishing points, V_R , V_L , and V_D are those formed by parallel lines in the direction of the axes of the object. Depending on the angle of the image plane to the object, one or two of these points may lie at infinity. The number of main vanishing points of an object (not at infinity) defines the order of perspective of the object. The object in Figure 7.5 has three orders of perspective, the points V_R , V_L and V_D being parallel to the X, Y and Z axis of the object respectively. Extending lines, referred to as vanishing lines, that are parallel to the object axes until they intersect forms these vanishing points. A line between V_R and V_L forms the true horizon of the image.

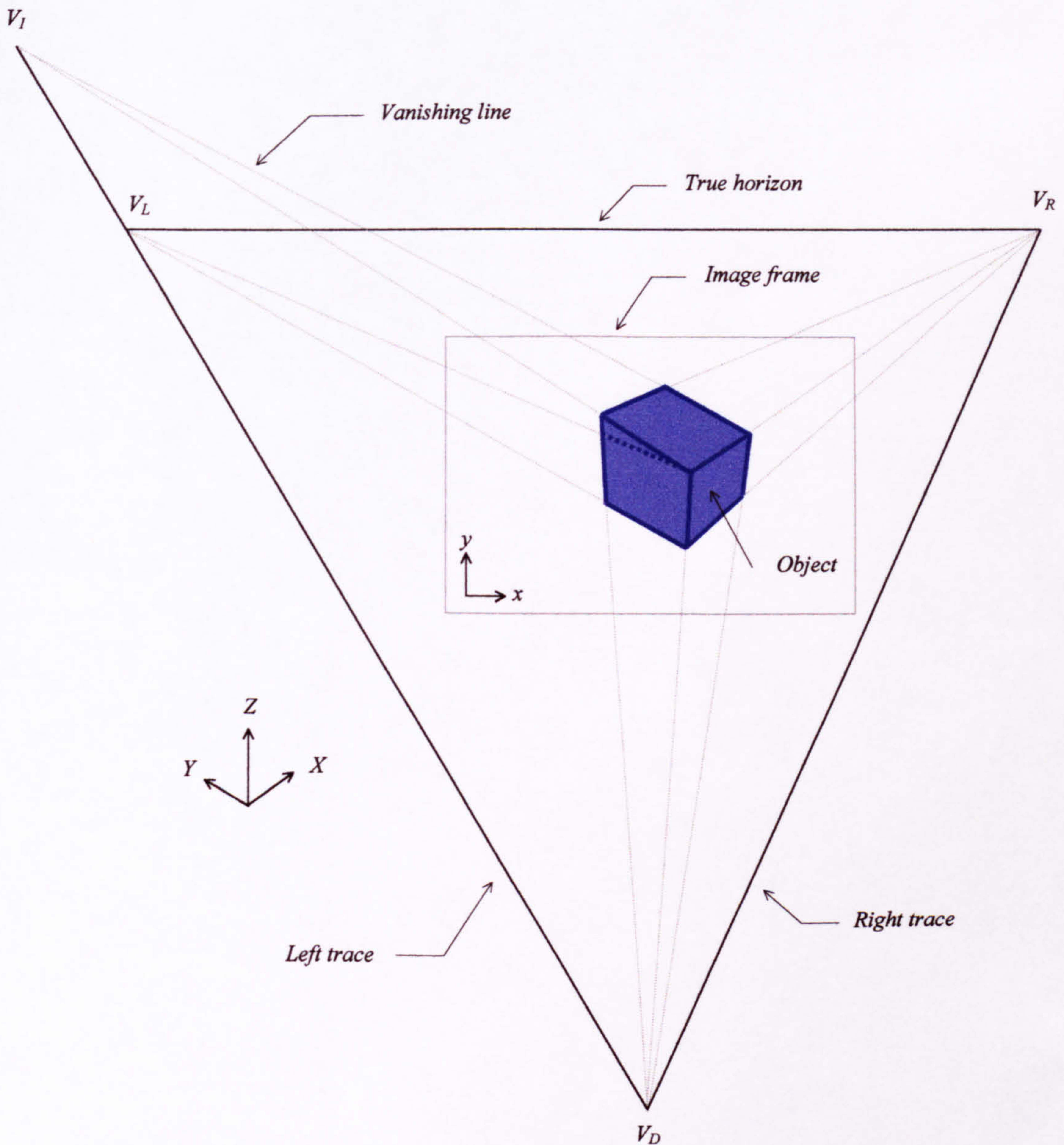


Figure 7.5: Location of vanishing points of an object

Figure 7.5. The three main vanishing points of an object V_R , V_L , and V_D are constructed by extending lines parallel to the object axes in the image plane until they intersect. Lines at an angle to the object axes intersect along the vanishing traces between the three main vanishing points.

For an inclined line on a vertical face, the vanishing point, V_L , will be on an extension of the left or right trace. For the case of an inclined line being on a horizontal face, the vanishing point will fall along the true horizon. These factors can be used to improve the position of vanishing points constructed graphically.

7.4 Graphical Techniques

This chapter discusses techniques that have an application for the quick and cheap collection of data for the production of architectural models for virtual worlds, or for where circumstances only allow a single photograph to be taken. Debevec *et al.* (1996, page 11) states that

“... computer models of architectural scenes have been especially popular subjects, mirroring the real-world popularity of architectural sites as places to visit. From this there is a call for a method to conveniently acquire photo-realistic models of existing architecture.”

As stated earlier, the majority of work done with single images imposes certain restrictions on the image. The main restriction is that the object has to be a two-dimensional plane, for example a wall, although this can be rotated in the three-dimensional world. The simplest and cheapest method involves the usage of a paper strip. For this method, a minimum of four points has to be identified on the image, for which object co-ordinates are known. To position a new point, lines are drawn between each point on the map and between each point and the new point on the image. The paper strip is then placed on the image and the intersections with the lines from one of the known points are then marked. The strip is then aligned on the map so that the marks coincide with the lines from the known point, and the position of the line to the new point is marked. If this is done for two of the known points then, where the lines from the known points through their relative positions for the new point intersect, is the position on the map of the new point (Doneus 1996).

A more common approach today is to rectify the image numerically using a CAD system or similar software, although it is possible to rectify the image optically. For

numerical rectification, a minimum of four known points on the plane to be rectified is required. These points are then indicated on the image, and their full three-dimensional co-ordinates entered. From these, it is then possible to calculate the relative transformations for the conversion between the two co-ordinate systems, image, and object. More advanced systems can allow the image to be rectified pixel by pixel into the three-dimensional co-ordinate system. This allows for the creation of textures for object planes to be made from the oblique photographs. The basis of more advanced graphical techniques to extract information from an image are the reverse of techniques originally used to draw perspective images.

The techniques explained here are based on those from the Canadian Department of National Defence (1988). Throughout the derivations, various symbols are commonly used to represent points and distances in the calculations

- x, y, z - co-ordinate system of the image
- X, Y, Z - co-ordinate system of the model
- $V_L, V_R,$ and V_D - main vanishing points of the model
- O - focal point of the camera
- c - relative focal length
- p - principal centre of the image
- $\omega, \varphi,$ and κ - angles of rotation about the x, y and z axes respectively

As stated earlier in Section 7.2, the vanishing points, V_R , V_L , and V_D of an object are a result of the relative camera position. The camera position can therefore be geometrically reconstructed from these points, see Figure 7.6.

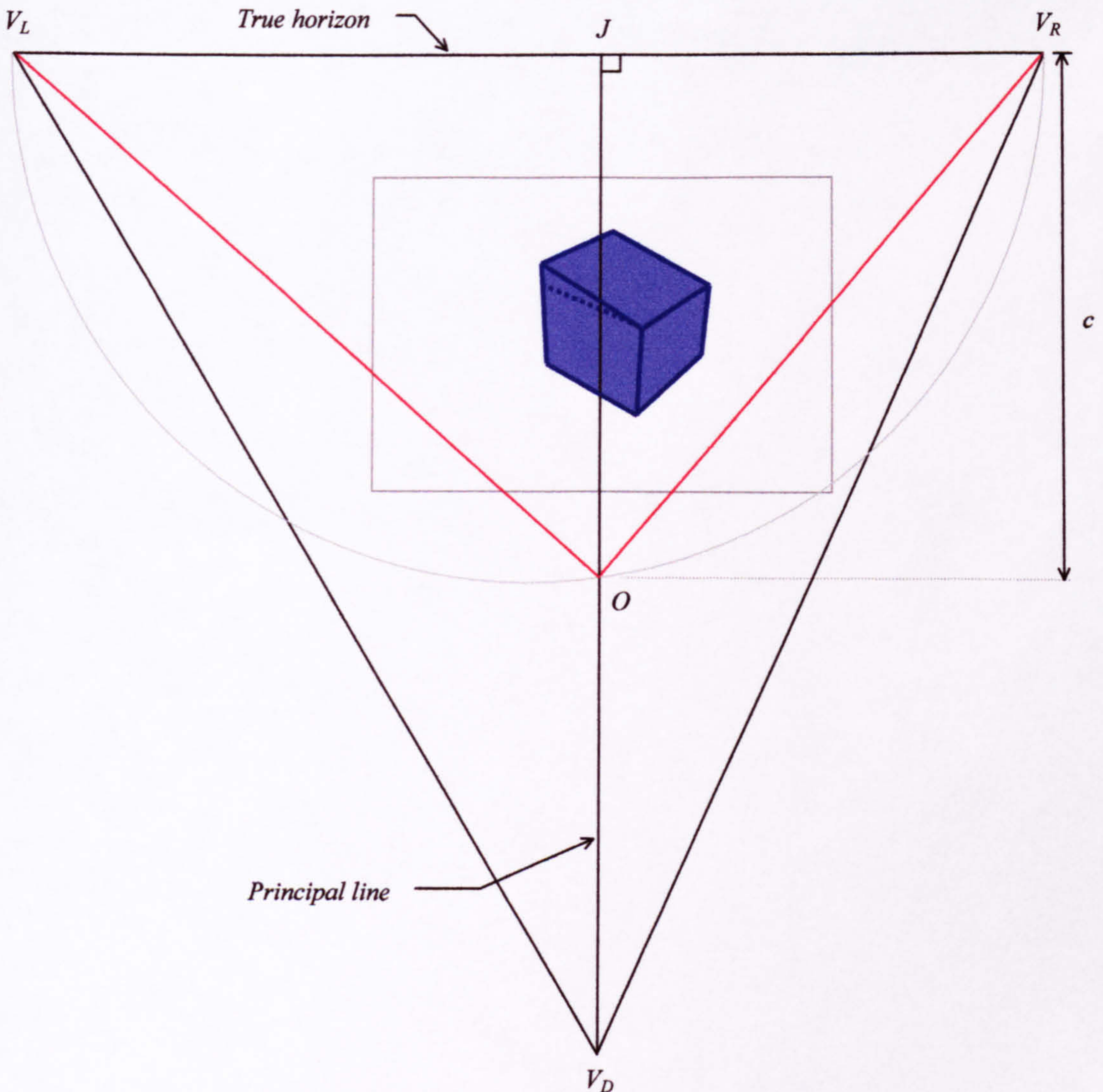


Figure 7.6: Three point perspective geometry

Figure 7.6. Where a semi-circle drawn between V_L and V_R intersects with the principal line is the relative position of the focal point. The distance from the focal point to the true horizon represents the relative focal length of the image. Lines drawn from the focal point to V_L and V_R are then in the direction of the object's X and Y axes, in plan view.

The example followed here details how to create a model from a full frame image with three orders of perspective. It is possible to use similar techniques for images with only two orders of perspective. For this case, the point V_D is found at infinity and therefore all verticals are drawn perpendicular to the true horizon. The first stage is to construct the relative position of the focal point, O , at the time of exposure.

The intersection of the principal line between V_D and J and a semi-circle drawn between V_R and V_L determines the position of the focal point, O , which in this case is the lens of the camera. The distance OJ is the relative focal length, c , of the camera for the image. The line OV_R will be in the direction of the X-axis of the object, similarly OV_L will give the Y-axis, for the plan view.

The Canadian Department of National Defence (1988) state that for an image where the tilt is less than five degrees, it can be treated as if it has perspective, and only gives a one-percent error in the resulting position. This is the case for many of the buildings used in this research. As the position of V_D is hard to locate accurately, as lines in its direction are near vertical, and therefore may not intersect cleanly, two-point perspective is commonly used for the graphical technique.

The generation of a plan view of the object can then be created from the image, as illustrated in Figure 7.7. Perspective verticals, lines drawn from the vanishing point V_D through points on the image, are drawn through g , h and i to intersect with the true horizon giving g' , h' and i' . The next stage is that at g' lines parallel to the X and Y axes of the object, in plan, are drawn. Visual rays constructed from O through h' and i' are then drawn to intersect with the axes from g' to give the plan view of the object.

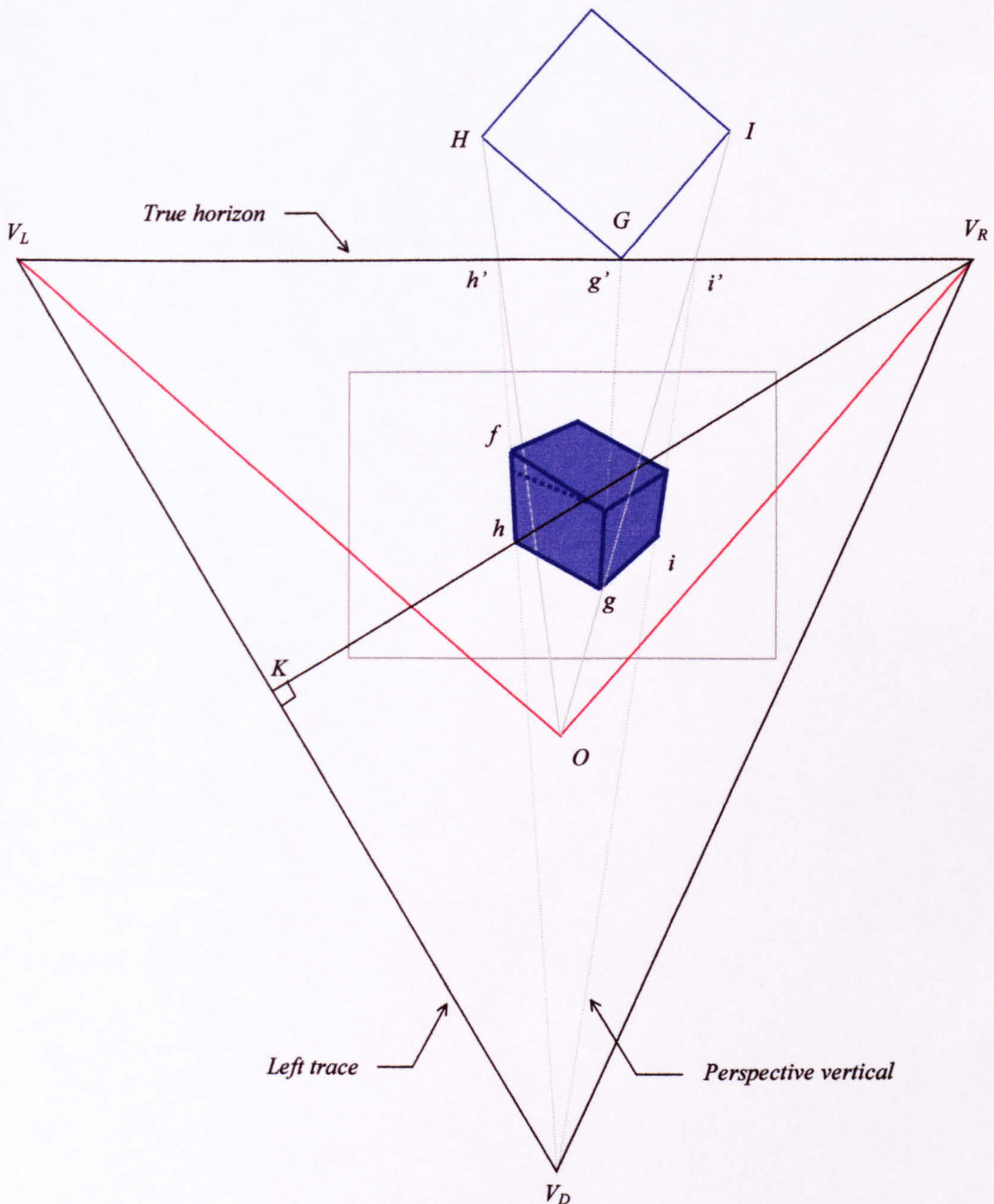


Figure 7.7: Generation of plan view

Figure 7.7. Perspective verticals are drawn from V_D through points on the base of the object to intersect with the true horizon giving h' , g' , and i' . Visual rays are then constructed from the focal point through the intersections. By then constructing lines parallel to the object axes from g' the plan view of the model can be created.

A similar approach is then followed to create the side elevation of the object (Figure 7.8). The relative focal length, c , of the camera is measured from K towards V_R . Then lines are constructed as before, through the points of interest to the left trace. Lines are then drawn from the new camera position through these points.

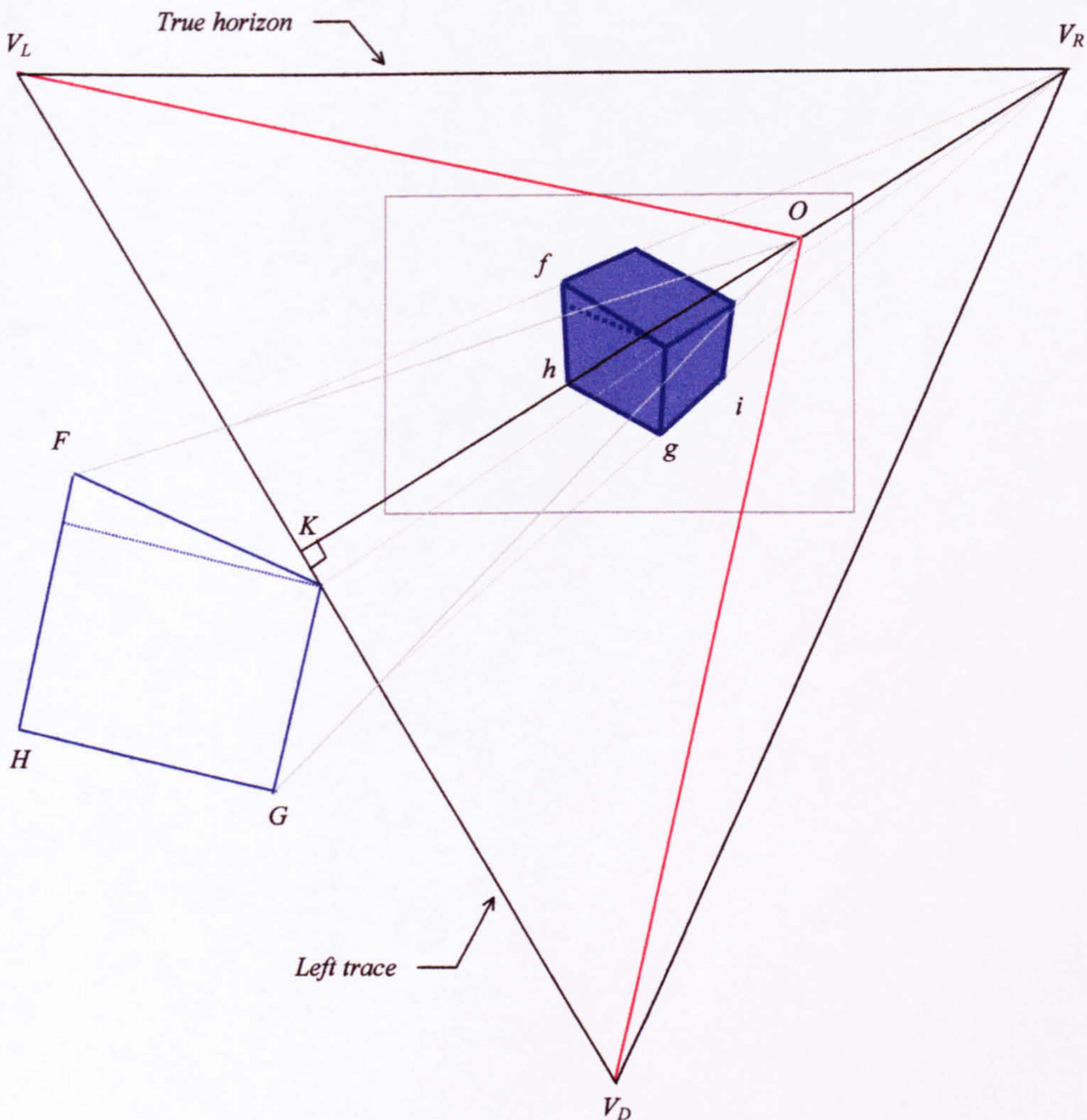


Figure 7.8: Generation of elevation

Figure 7.8. By following a similar approach to the construction of the plan view, the elevations of the model can also be created.

From these views of the model measurements can be taken to construct the final three-dimensional model. Similar techniques can be employed for the generation of other views and to allow the usage of complex shapes.

7.5 Automated methods

The artificial intelligence community has carried out research into the extraction of information about an object from a single image. Here the aim is for an entirely automated method for applications such as machine vision. Although some methods

follow a similar approach to that discussed above, the usage of shadows, textures, range images, labelled line drawings and object matching are also examined.

Much of the work carried out in the artificial intelligence community is concerned with the reconstruction of Legoland scenes. The Legoland world idea represents a world where physical edges are only oriented in the direction of the three orthogonal directions, unlike the Lego of today. This simplifies the problem domain and is true for most manufactured objects.

7.5.1 Labelled line drawings

A two-dimensional line drawing can be converted into various three-dimensional Legoland objects, depending on how the junctions and edges of the drawing are labelled. It is possible to define junctions into four types; L, T, Y and E. Parodi and Torre (1994) define these as

- if a junction has only two incident segments, it is classified as an L
- if two of the three incident segments are collinear, it is classified as a T, the two collinear segments are referred to as the head of the T, the remaining segment as the foot
- if all the angles between the segments are smaller than 180° the junction is classified as an Y, otherwise as an E

Once the junctions have been identified and labelled it is then necessary to label the edges, and this is done by the usage of +, - and \rightarrow symbols, which characterise the three-dimensional properties of the edge it represents. An edge labelled as + corresponds to a convex edge, and similarly - refers to a concave edge. The \rightarrow symbol is associated to a convex edge in which one face occludes the other (Clowes 1971,

Hoffman 1971). An example of this technique of labelling a complicated shape can be seen in Figure 7.9.

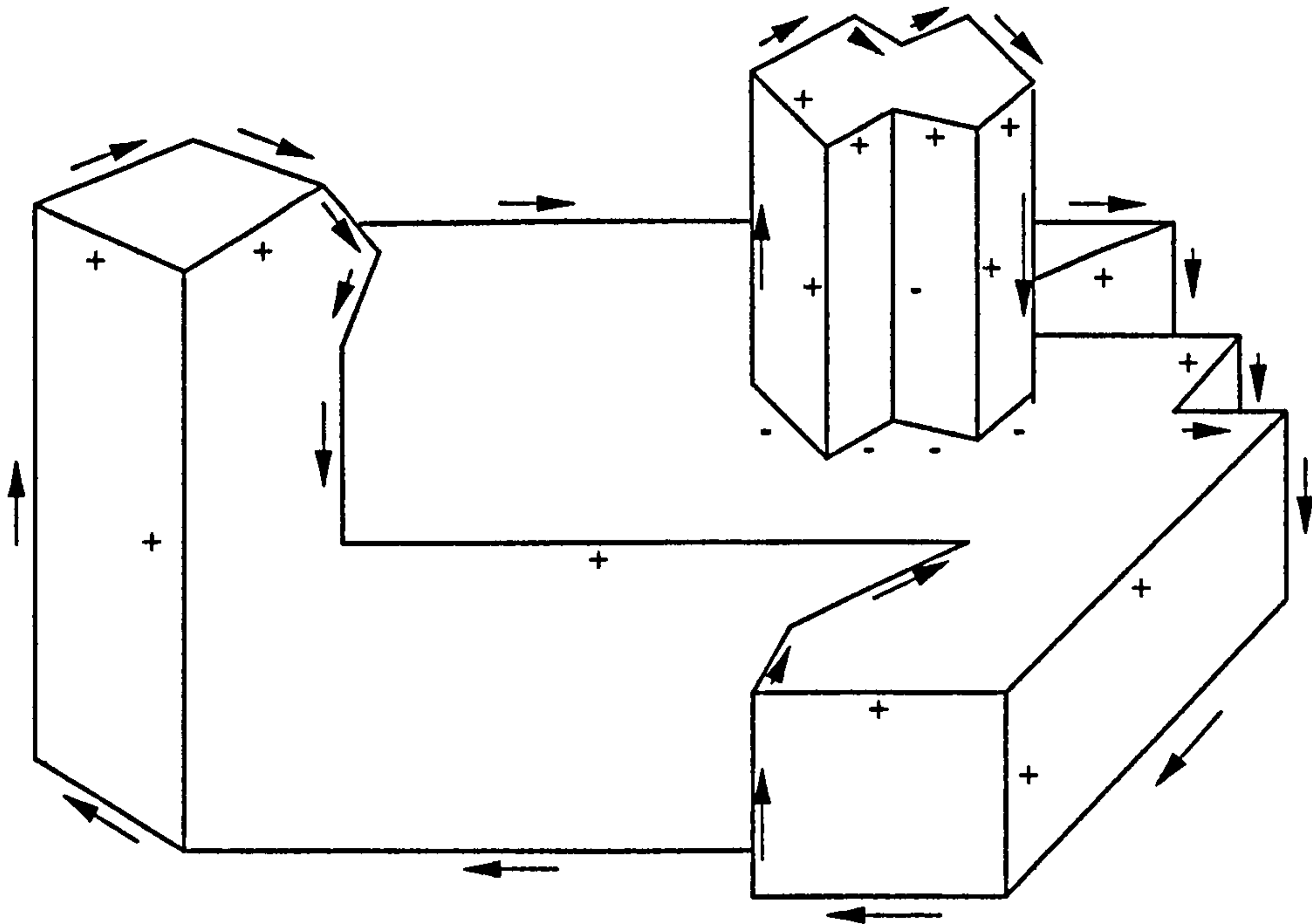


Figure 7.9: An example of a labelled line drawing

Figure 7.9. The three-dimensional shape of an object can be extracted from a two-dimensional labelled line drawing. An edge labelled as + or - corresponds to a convex or concave edge respectively. The \rightarrow symbol is associated to a convex edge in which one face occludes the other (Sparodi and Torre 1994).

This is not an entirely automated process, as there is a requirement on a user to identify and label the drawing. Once the line drawing has been completely labelled then, automated methods can be used to interpret the drawing as discussed by Kanade (1981).

7.5.2 Gaussian Mapping

The automatic extraction of vanishing points can be aided by the usage of Gaussian mapping. Gaussian mapping transforms vectors in three-dimensional space into points on a unit sphere centred at the origin of the object co-ordinate system. Points on the surface of the sphere can be used to represent the orientation of lines, where the vectors represent direction cosines, or planes, where interpretation of the vectors is as planar

normals. Any point on the sphere can therefore represent a family of parallel vectors in the world (Barnard 1983).

Lines on the image, which are projected onto the Gaussian sphere, form great circles. The intersection of these great circles on the sphere corresponds to the intersection of the lines in the image. Therefore, to find the vanishing points, lines in the image first are projected onto the sphere to form great circles. The points on the sphere where several of these great circles intersect are therefore the vanishing points. A further method is to use the duals of the vanishing points. Barnard (1983) describes the dual of a vanishing point as a vanishing line, in this case a great circle on the sphere. Therefore for a vertical vanishing point, the dual is the horizon line and similarly if two horizontal vanishing points are found, their duals intersect at the vertical vanishing point. This allows the creation of vanishing points in images where it is not possible to identify either horizontal or vertical segments.

7.6 Summary

As shown in this chapter various techniques can be used to gain information about the size and shape of a three-dimensional object from an image. By gaining a greater understanding of the mathematics of projection and how this relates to what is seen, complex three-dimensional models can be reverse engineered from single images.

The failure of the automatic methods is often related to the problems of the automatic generation of vanishing points and the extraction of the outline of an object from the image. As shown with the optical illusions it is not always safe to assume that what is seen can exist. For this reason, a fully automated technique is not used in this research; instead, user input is used for this information. It is recognised though that in the

generation of an automated process, these techniques could supply the information currently obtained from the user.

For simple models, the graphical solutions explained here can be used to construct plan and elevation views of three-dimensional models. What is more common with geographic information is the requirement to gain data in a digital form that can then be combined with existing information to produce virtual worlds. The next chapter examines the creation of a mathematical approach to the single image problem, building on the ideas described here.

8. Phoenix Algorithms

This chapter discusses the mathematical solution to the single image problem explained in Chapter 7. The mathematical solution allows for the technique to be incorporated into a stand-alone application, or as an addition to other software, such as a CAD application. This makes the process faster and more robust than the use of the graphical techniques explained previously. Parts of the approach can also then begin to be automated to aid in the interpretation of an image. As the model is created digitally, then the ability to transfer it to CAD or VR packages becomes available. Also, rectified textures can be extracted from the image and used in the final model.

The first algorithm (Parsley and Taylor 1998) was developed before the graphical solutions, detailed in Section 7.4, were made accessible by the Canadian Government. Therefore, this follows a more photogrammetric approach to recreating the models from the image. The second version of the algorithms builds on the graphical methods and further research into the area to enhance the technique. Both are explained in this chapter to illustrate the background to the process.

The technique has become known as ‘Phoenix’, after the mythical bird that rose from the ashes. It is used to illustrate the fact that the software allows buildings that are now destroyed to rise again. However, there is no reason why it cannot be used on buildings or other objects that still exist.

8.1 Background

The background research to Phoenix is from the usage of geometric constraints in the generation of an object from its perspective image carried out by the artificial

intelligence community. This allows the unknowns in the photogrammetric solution of the problem to be constrained.

SolidFit (Newsam 1996) requires the user to indicate on the image geometric relationships between points, lines and faces. These may be the colinearity of three points, that four points make a rectangle, or that the angle between sets of lines is equal. From these, the software then produces a best-fit model using a least-squares approach to minimise weighted errors. Similarly to this, the Façade software (Debevec *et al.* 1996), uses a combination of geometric primitives for example cubes, spheres, and cylinders, with geometric constraints for example symmetry, 'next to' and 'on top of', to generate a basic model. The model position and rotation is then constrained by the relationship of these objects to the image before a final model is created. Following these approaches Phoenix uses a combined photogrammetric and geometric approach to obtain a solution. The initial orientation of the camera in relation to the object is generated, then by applying simple constraints on the three-dimensional position of points, a model is built up.

8.1.1 Kodak Dimension, PhotoModeler and Canoma

Towards the end of this research, various commercial products were beginning to appear on the market that could perform a similar functionality to that given by the usage of the Phoenix algorithms. These are PhotoModeler, Kodak Dimension, and MetaCreations Canoma. Unfortunately, there was no opportunity to directly compare the functionality of these products, in single image mode, to that given by the Phoenix algorithms. The products use various methods to obtain the final model from a single image. PhotoModeler (discussed further in Section 9.1.3) and Dimension follow a

similar technique to that described here for Phoenix, while Canoma uses the position of regular geometric shapes to fix the model.

The Dimension software available from Kodak is only available for use by governmental bodies. It requires the user to decide the level of perspective (either one-, two-, or three-point) that the image they are using has, before they begin to create the model. At the time no information could be gained as to the actual algorithmic approaches used by the software, but the promotional literature suggests that it has been developed using information available from the military on the solution of single images.

Canoma from MetaCreations allows the user to generate the model by placing regular geometric shapes on to the image in line with objects, for example, boxes around buildings. The user then builds up the model they wish to create by associating relationships between the geometric objects and a ground plane. Once the model has been created, textures can be created for the model from the supplied image.

8.1.2 Mathematical notation

For the following equations, the subsequent mathematical notation is used to represent points, distances, angles, and co-ordinates

- aJ - the distance between two points; e.g. the distance from a to J
- JOV_D - the from-at-to angle; e.g. the angle at O from J to V_D
- x_a - the x value of a point in the image co-ordinate system; e.g. point a
- xV_D - the x value of a vanishing point in the image co-ordinate system; e.g. the V_D vanishing point
- X_F - the X value of a point in the object co-ordinate system; e.g. point F

This notation is preserved for both examples of the Phoenix algorithms. Where possible labels for points and distances present in each of the approaches have remained the same to aid clarity. During the calculations, the two co-ordinate systems, those of the image and the object, have the same unit distance. Once the model is created, it can be scaled up to the correct real-world dimensions if required.

8.2 Principal Point

The mathematical solution of the camera position requires the identification of the location of the principal point, p , of the image. This point is formed by extending a line perpendicular to the image plane, through the focal point of the camera, onto the object. The principal point is important, as it shows the point on the object that the camera was pointing at during exposure. The mathematical approach uses a line through this point and perpendicular to the image plane as the z-axis of the image co-ordinate system. This is required to calculate the relative rotations between the image system and the object system.

For calibrated cameras, the principal point is either marked on the image, or found by joining the fiducial marks around the edge of the image. In the research, the aim was to allow for the use of uncalibrated cameras for the purposes of model creation. Therefore, it is necessary to be able to calculate the principal point from information contained in the image itself, as the camera may no longer exist.

For the case of an uncalibrated camera, on a full-frame image (or a negative), the principal point is commonly the centre point of the image, if lens distortion is ignored. The difficulty in calculating the exact position of the principal point occurs on an image where the edges of the image are not defined, most likely due to the image being

cropped (Figure 8.1). Here an approach using the properties of the objects in the images must be used, as this is the only information available. This approach can also be used for full-frame images where a more accurate position of the principal point is required.

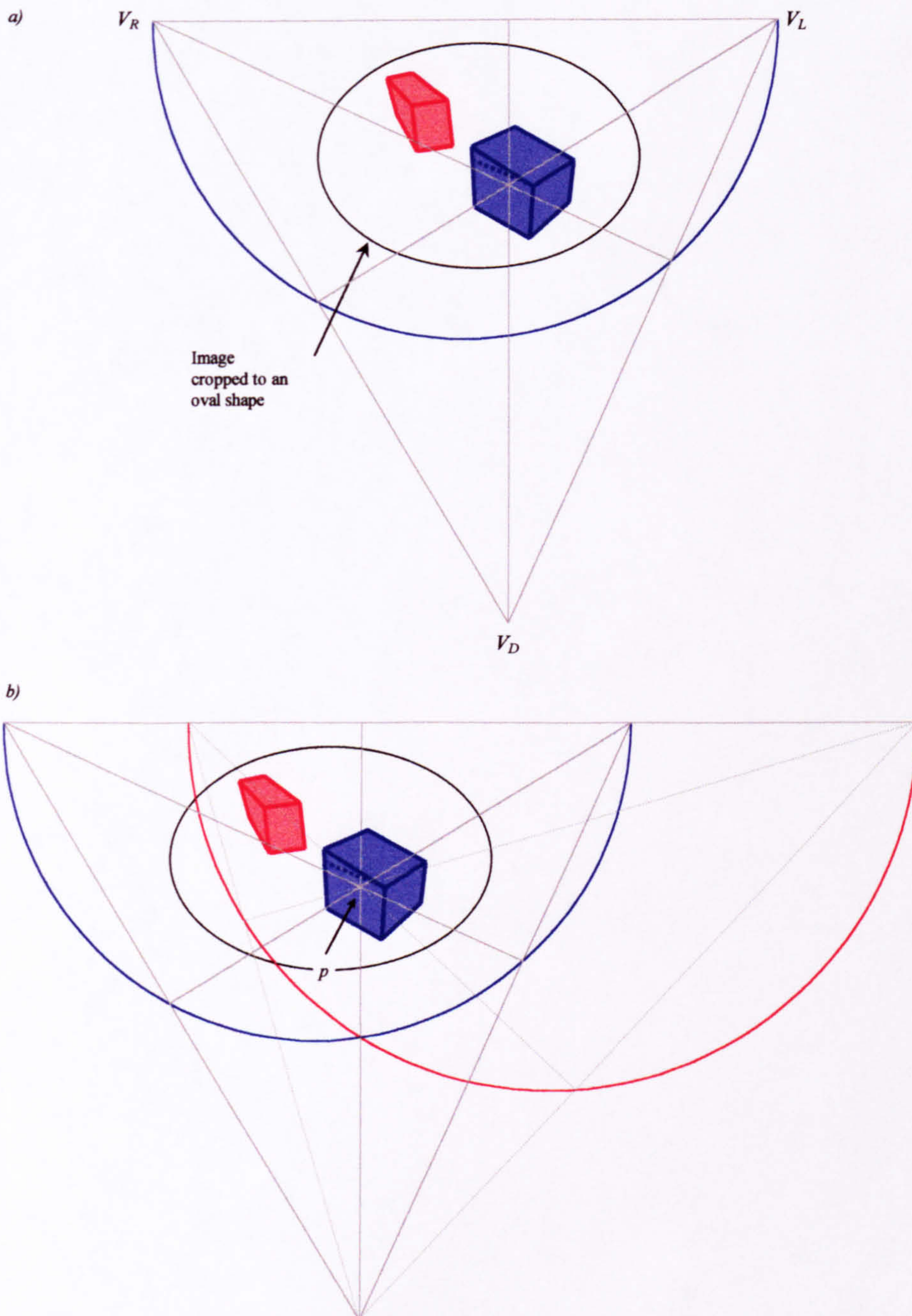


Figure 8.1: Graphical creation of the principal point, p for a cropped image

Figure 8.1. The principal point, p , of a cropped image can be created by discovering the epicentre of the perspective triangles surrounding the objects in the image. If multiple objects at different rotations to the camera are used as in (b) then a more precise location can be found.

The cropped image solution relies on the fact that objects at different rotations to the camera will have different vanishing points for their axes along the true horizon line. These therefore make differing perspective triangles around the objects when lines are drawn between them. The orthocentre of each perspective triangle is defined as the point where perpendicular lines from each edge of the triangle through the opposite corner intersect. Due to the properties of the perspective projection, this is also the location of the principal point, p , of the camera. As the camera position is constant for each of the triangles, then if the vanishing points are located accurately then the orthocentres of all the triangles will intersect at the same location. Therefore, just one perspective triangle can be used to generate the principal point (as illustrated in part (a) of Figure 8.1), although the accuracy can be increased by selecting the average of multiple orthocentres (part (b) of Figure 8.1).

It is also possible to gain directly a solution for the relative focal point, O , of the camera. This will be the point where the semi-circle arcs between the sets of vanishing points intersect. This method is an alternative graphical means for finding the location of O to that mentioned earlier in Section 7.4.

8.3 Phoenix v1

The basis for the original version of Phoenix is an adaptation of the techniques used in photogrammetry. The approach used is to rectify the image to a vertical plane and then convert the rectified points to object co-ordinates. For the Phoenix v1 approach, the user is required to indicate a known angle between two vertical faces of the object on the image.

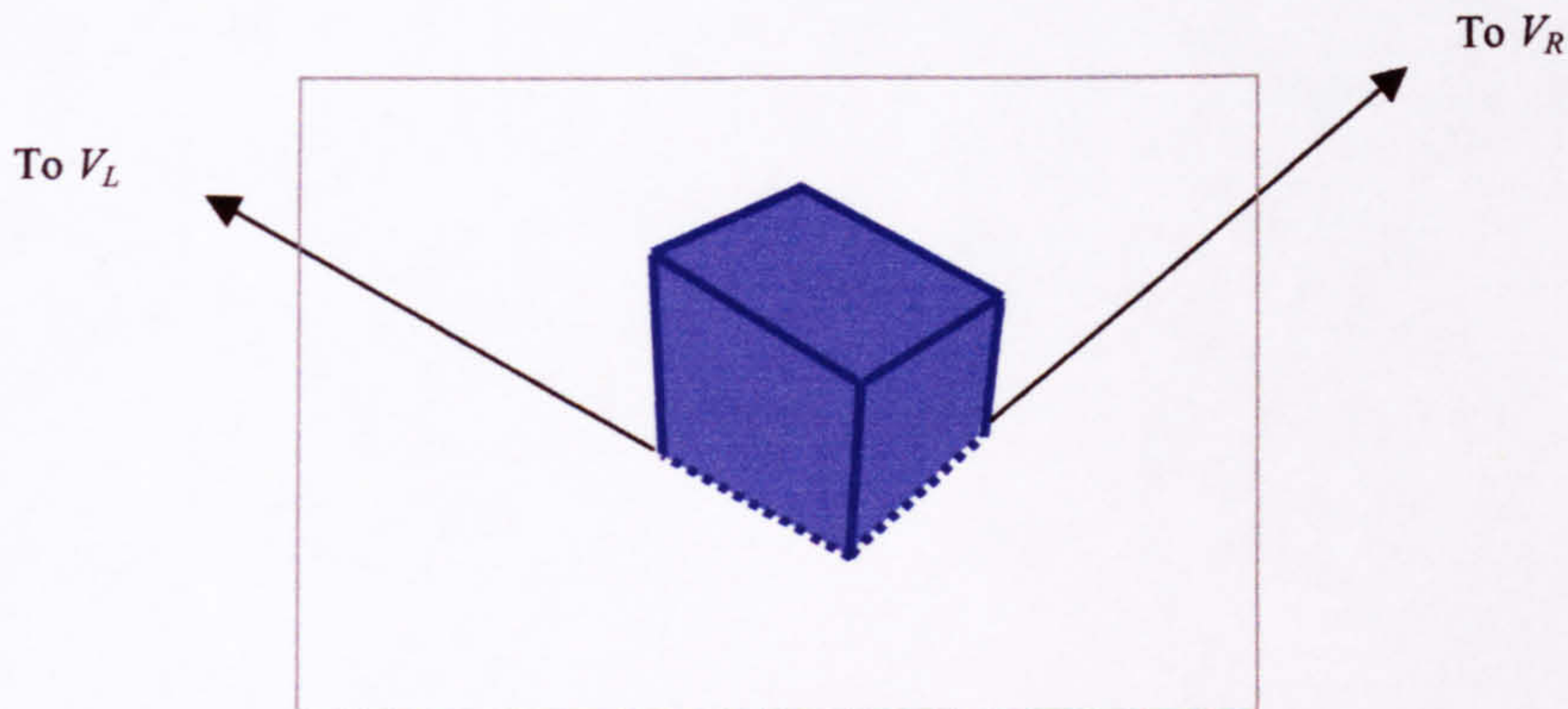


Figure 8.2: User input for Phoenix v1

Figure 8.2. For the initial version of the Phoenix algorithms, the user was required to supply the positions of V_L and V_R as detailed earlier and two lines for which the internal horizontal angle is known.

This is done by the user indicating two horizontal lines on the object in the direction of the planes, as shown dotted in Figure 8.2. The user is then also required to supply the horizon line of the object by indicating horizontal parallel lines on the object. The approach works by calculating various values for the relative focal length and angle of tilt of the image. From this the three-dimensional co-ordinates for the lines indicated on the image can be calculated. When the angle between these matches, within a tolerance, specified by the user, then the relative focal distance and tilt of the camera is fixed, allowing the tilted image to be rectified to a vertical plane. The generation of further points is then possible by calculating the intersection of lines from the focal centre, through a point on the rectified image to a plane defined in the model space.

Figure 8.3 shows a side view of the geometry of a tilted image as used for the Phoenix v1 algorithms. The point O is the focal point of the camera at the time of exposure. The vanishing points, V_R , and V_L (discussed in Section 7.3) for the object and the focal point of the camera O , generate the horizon plane. The point J can then be defined as the point where a vertical line on the image plane through the principal point, p , intersects the horizon plane. The angle γ is defined as the vertical angle between the

horizon plane and the line Op . The point where a line from O bisecting the angle γ intersects the vertical line from p is called i and falls on the horizontal line of intersection between the image plane and the vertical plane. The point J' is vertically above i where the vertical and horizon planes intersect. Two left handed co-ordinate systems are then set up such that p is used as the origin of the image co-ordinate system, while the point J' is used as the origin of the object co-ordinate system.

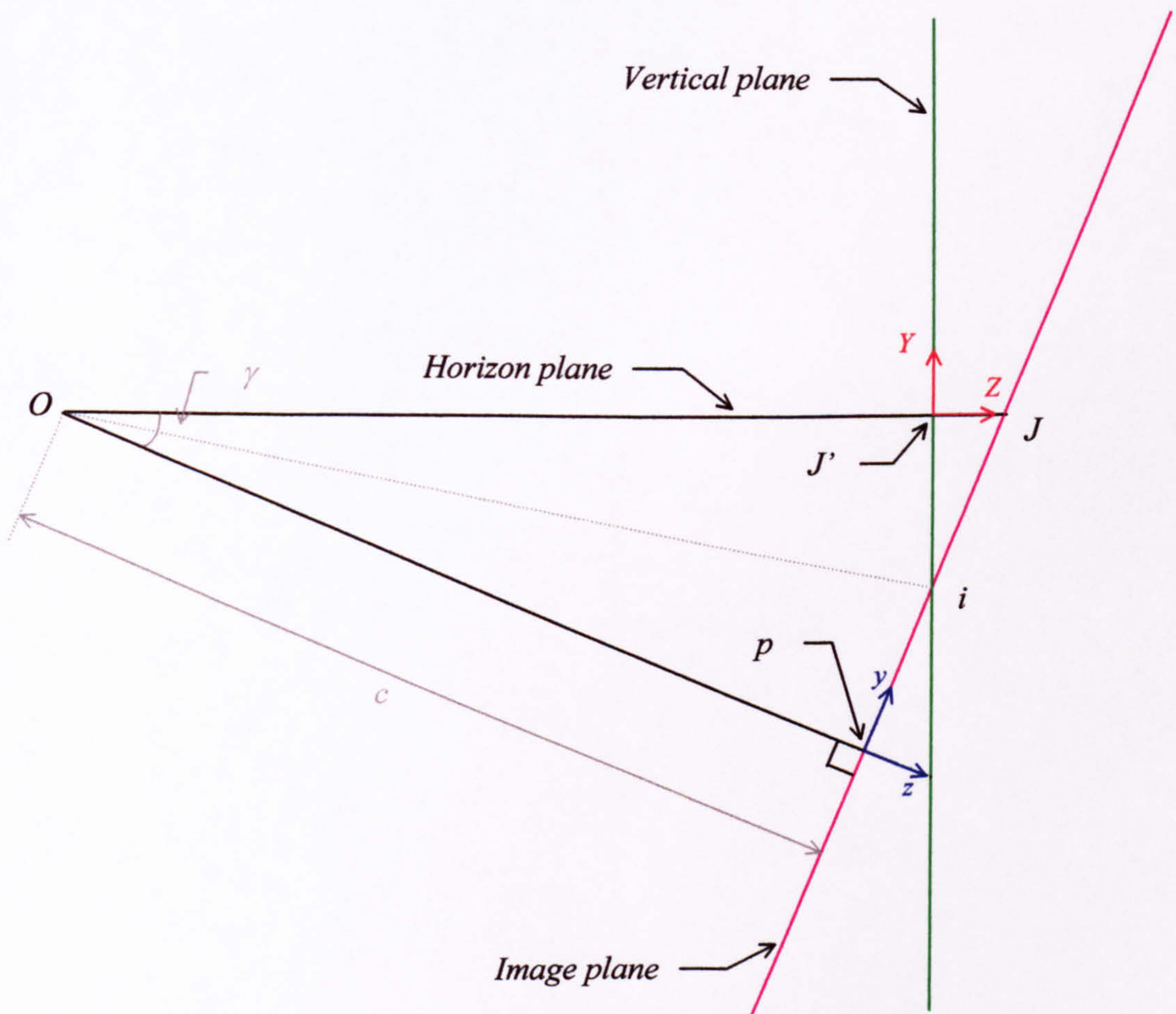


Figure 8.3: Geometry for Phoenix v1

Figure 8.3. The tilted image is first rectified to a vertical plane before the image co-ordinates are converted to model co-ordinates.

The unknowns in Figure 8.3, which require calculation, are therefore the relative focal length, c , (between O and p) and the angle of tilt, γ (between the horizon plane and the line Op). For a given focal length, c , the angle of tilt, γ , can be shown to be

$$\gamma = \tan^{-1}\left(\frac{pJ}{c}\right)$$

Equation 8.1

Here pJ represents the distance from the point p to the point J in the unit of the image co-ordinate system. The angle of tilt, γ , can be discovered by using an iterative approach to discover the value of c that subtends the internal angle stated by the user. This internal angle can be calculated from the information in Figure 8.3 and Equation 8.1 using the equations below.

The isocentre i can be seen to have, in the image co-ordinate system, the co-ordinates

$$\begin{aligned} x_i &= 0 \\ y_i &= \tan\left(\frac{\gamma}{2}\right) \end{aligned}$$

Equation 8.2

With this information known it is possible to use photogrammetric equations to correct for the tilt. This involves projecting the co-ordinates of points from the tilted image on to a vertical plane. This is done in two stages, firstly the radial displacement, Δ , along a line from the isocentre to the point on the tilted image is calculated by

$$\Delta = \frac{(r \cos^2 \beta \sin \gamma)}{(c - r \cos \beta \sin \gamma)}$$

Equation 8.3

The value β is the clockwise angle from the principal line, and r is the radial distance from the isocentre to the point. Once Δ is known it can be resolved in to its x- and y-axis parts by use of trigonometry.

Hearn and Baker (1986) show that the point, f , where a line from the focal centre O to a point on the object F intersects with the vertical plane can be expressed in a parametric form as

$$\begin{aligned}x_f &= X_F - \delta X_F \\y_f &= Y_F - \delta Y_F \\z_f &= Z_F - \delta(Z_F + c)\end{aligned}$$

Equation 8.4

The parameter δ can take a value from zero to one, and represents the position of the vertical plane in relation to the distance from the model to the focal centre. When δ is zero, then Equation 8.4 yields the point F , the point on the model. Similarly, when δ is one the result is the focal centre, O , which has the co-ordinates $(0, 0, -c)$ in both co-ordinate systems. The point f therefore lies on the image plane when z_f is set equal to zero. The value of δ can be resolved by rearranging the final part of Equation 8.4 to give

$$\delta = Z_F / (Z_F + c)$$

Equation 8.5

This allows co-ordinates for the lines indicated by the user to be created in the object co-ordinate system, and therefore the angle between these two lines can then be determined. As stated earlier, if this is not within the stated tolerance, then the value of c is increased by a given amount, and the process is repeated from Equation 8.1. To improve the final accuracy, but keep calculation speed, the value of the increment to c can be made to vary dependent on the difference between the calculated value and the stated value.

Once within the tolerance, vertical planes are generated to represent the walls initially indicated by the user. Further detail can then be picked up on these planes by calculating where a line from the focal centre through the rectified image intersects with a selected plane using Equation 8.4. Further information can be added by restricting one of the object axes to a value, and using this or other geometric constraints to solve the value of δ .

8.4 Phoenix v2

In the revised Phoenix algorithms, a more automated approach to the solution of the graphical problem, discussed in Section 7.4, is followed. This solution removes the iteration of a value for the relative focal length, c , used in the approach above. The removal of the iteration process has two effects on the method. Firstly, the technique is speeded up, as c is gained from a direct calculation, and secondly the position of the focal centre is fixed to a point, rather than falling between a tolerance range. Therefore, the method will give increased accuracy over the previous calculation due to this removal of uncertainty.

As stated in Section 7.3, an object possesses one-, two- or three-point perspective. For the explanation detailed here, the case of three-point perspective is illustrated. For two-point perspective, it can be seen that V_D then falls at infinity, $(0, \infty)$ as lines drawn through vertical lines on the object towards V_D are parallel. General mathematical principles can be used to prove that two parallel lines will converge at $(0, \infty)$. If this co-ordinate for V_D is used in the following equations, a correct result will be given, as a number divided by ∞ gives a result of zero.

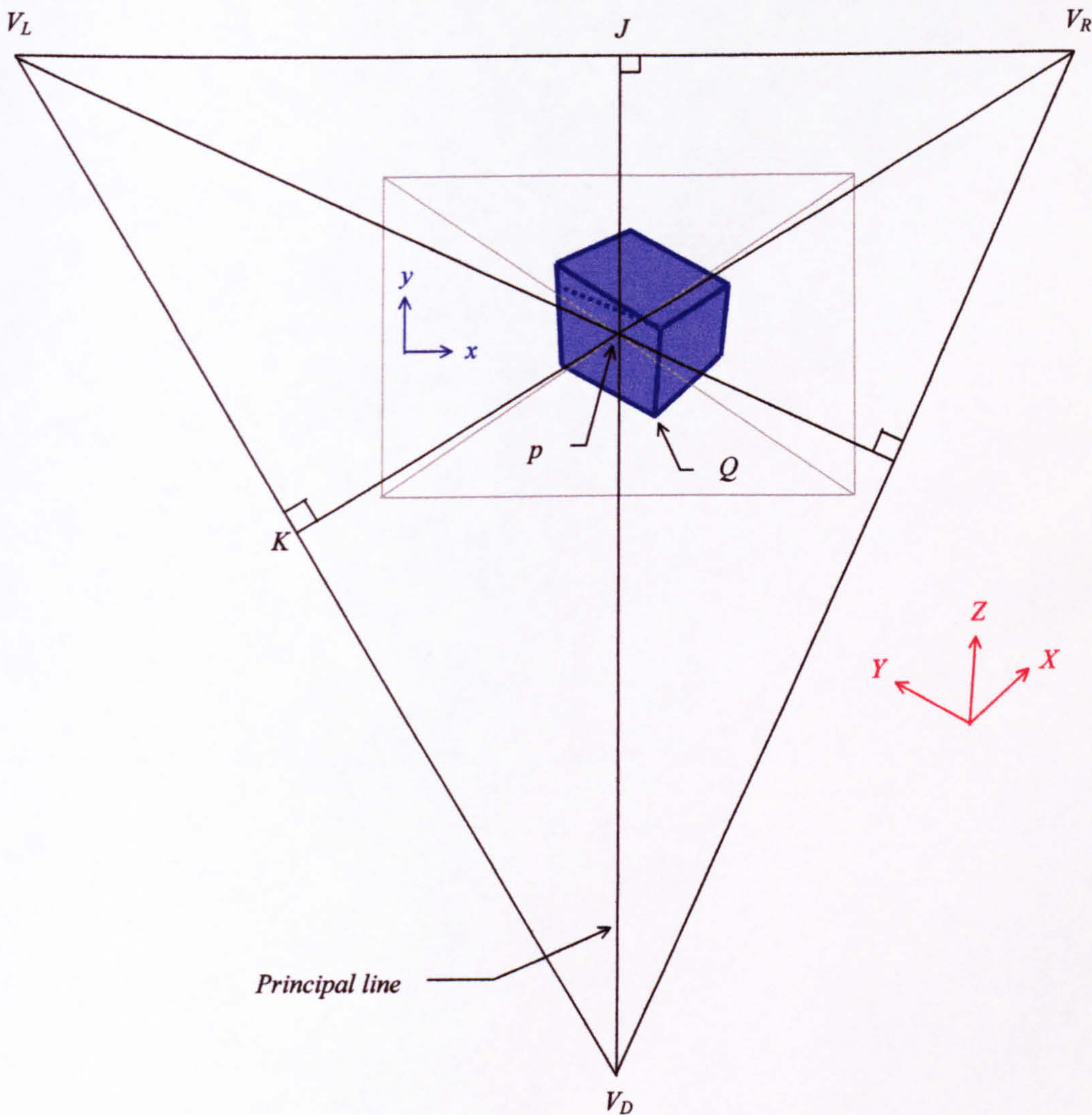


Figure 8.4: Perspective triangle

Figure 8.4. The three main vanishing points of an image form the perspective triangle around the image. The orthocentre of the triangle should coincide with the principal centre of the image. If this is not correct then the vanishing points are incorrectly positioned or the image has been cropped.

The three vanishing points V_R , V_L and V_D , form a triangle around the image which has its orthocentre at the principal centre of the image, p , see Figure 8.4 above. If the principal centre of the image and the observed centre of the image do not coincide then it implies the vanishing points are incorrectly calculated, or the image has been cropped. In this case, after checking the position of the vanishing points then the orthocentre of the perspective triangle should be taken to be the principal centre of the image. By observing the vanishing points of objects at various rotations to the image plane, it is

possible to enhance the position of p ; this was discussed in more detail in Section 8.2. The error produced by incorrectly positioning p is discussed further in Chapter 9.

For this solution, the co-ordinate systems are set up differently to that detailed in Section 8.3. The image co-ordinate system (ICS) is a left handed co-ordinate system. The origin is at the focal centre, O , with the axes labelled as x , y and z . The y -axis is defined as the line passing through the points, p , and V_D . The image plane is then a plane in the x , y -axes going through $(0, 0, c)$. Similarly, a right-handed object co-ordinate system (OCS) is set up with origin at Q , and axes labelled as X , Y and Z . The position of Q is decided by the user, though it is suggested to select a point where the axes of the object intersect, as shown in Figure 8.4.

The user is required to supply the principal centre of the image, p , the vanishing points, V_R and V_L (the point V_D is not used), and a point on the image that will be the origin for the OCS, Q . It is then possible to calculate all other values from these. The point V_D is not used in the calculation due to the difficulty in accurately constructing it graphically. This is because quite often the angle of tilt in the vertical plane between the camera and the object is relatively small. The main use of the point V_D is in finding the location of the point p if it is not available, for an example of this see Section 8.2.

With the two co-ordinate systems set up, the conversion between image and model is a defined as a translation and a series of rotations (Cooper and Robson 1996). This can be represented mathematically by

$$F = O - \mu R^T f$$

Equation 8.6

Here F and f refer to the same point as represented in the OCS and ICS respectively. The translation O is the position of the focal centre in relation to the origin of the OCS and R^T is a matrix of rotations. R^T is made up of three rotations, ω , φ and κ around the x, y and z axes respectively. Due to how the ICS and OCS are defined, there is no rotation around y, so φ is zero. The value of μ is a representation of the ratio between the distance between the origins of the two co-ordinate systems and the distance between the point on the object, and its related point on the image. Therefore, it is calculated by the geometric constraint put on the point being positioned, and varies between points. In full matrix notation the above equation can be represented as the following

$$\begin{bmatrix} X_F \\ Y_F \\ Z_F \end{bmatrix} = \begin{bmatrix} X_O \\ Y_O \\ Z_O \end{bmatrix} - \mu \begin{bmatrix} \cos \kappa & -\cos \omega \sin \kappa & \sin \omega \sin \kappa \\ \sin \kappa & \cos \omega \cos \kappa & -\sin \omega \cos \kappa \\ 0 & \sin \omega & \cos \omega \end{bmatrix} \begin{bmatrix} x_f \\ y_f \\ -c \end{bmatrix}$$

Equation 8.7

As an image, rather than a negative is being used, then the z-axis value is reversed to convert between a left-hand co-ordinate system and a right hand one. It is possible to use the same equations for a negative but it has to be remembered to remove this step of reversing the axis (therefore z_f will be represented by c). The six constant unknowns in this equation, which are initially calculated by Phoenix are the relative focal length, c , the position of the focal centre, (X_O, Y_O, Z_O) , and the two rotations, ω and κ . As stated above, the value of μ is calculated at a later stage for each point created in the model.

From the perspective triangle, shown earlier in Figure 8.4, the relative focal length of the image can be found. Zhao and Sun (1994) state that lines that are parallel to the X axis will intersect at the point V_R when they are projected in the perspective image

plane. Therefore, lines from the focal centre, O , in the V_R direction must be parallel to the X-axis because its perspective projection is also through the point V_R . For the same reason, lines from O to V_L and V_D must be parallel to the Y and Z-axes respectively. Thus, in object space, the lines OV_R , OV_L and OV_D are found to be perpendicular to each other and represent the direction of the object axes.

The point J is defined as the point where a line from O in the zy plane and parallel to the XY plane intersects with the line V_LV_R ; this can be seen in Figure 8.5. As a line from O to V_D is parallel to the Z-axis (as stated above), the angle JOV_D can therefore be seen to be 90° , which implies that the distance c can be expressed as

$$c = \sqrt{(pJ * pV_D)}$$

Equation 8.8

The position of V_D is not given by the user, therefore from the perspective triangle it can be seen that the distance c can be expressed in terms of the points K and V_R as;

$$c = \sqrt{(pK * pV_R)}$$

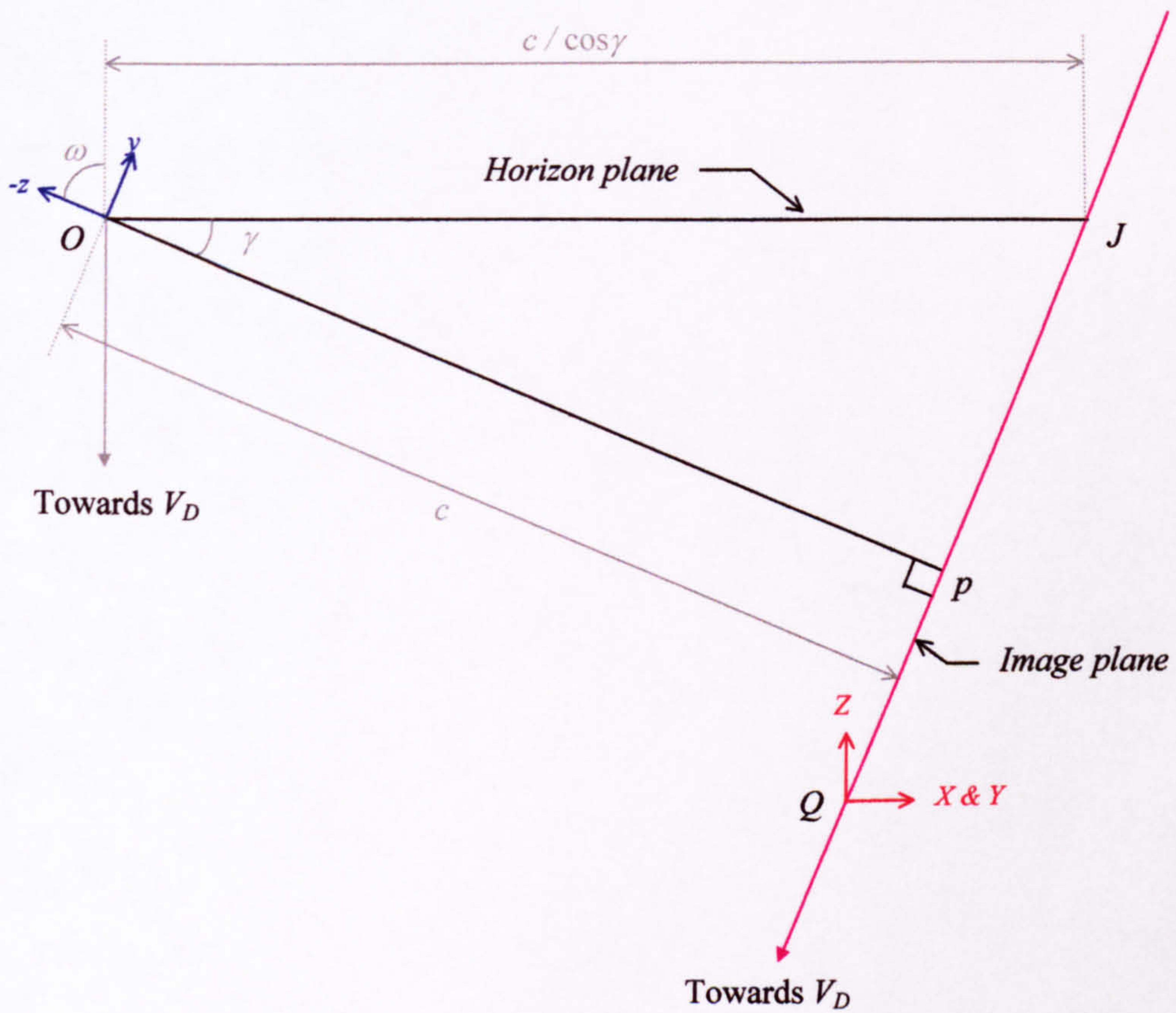
Equation 8.9

The distances pK and pV_R require the calculation of the distance pJ , which can be calculated from the intersection of the line from V_R to V_L with the principal line.

$$pJ = yV_R + \left[\frac{xV_R}{(xV_R - xV_L)} \right] * (yV_L - yV_R)$$

Equation 8.10

(a)



(b)

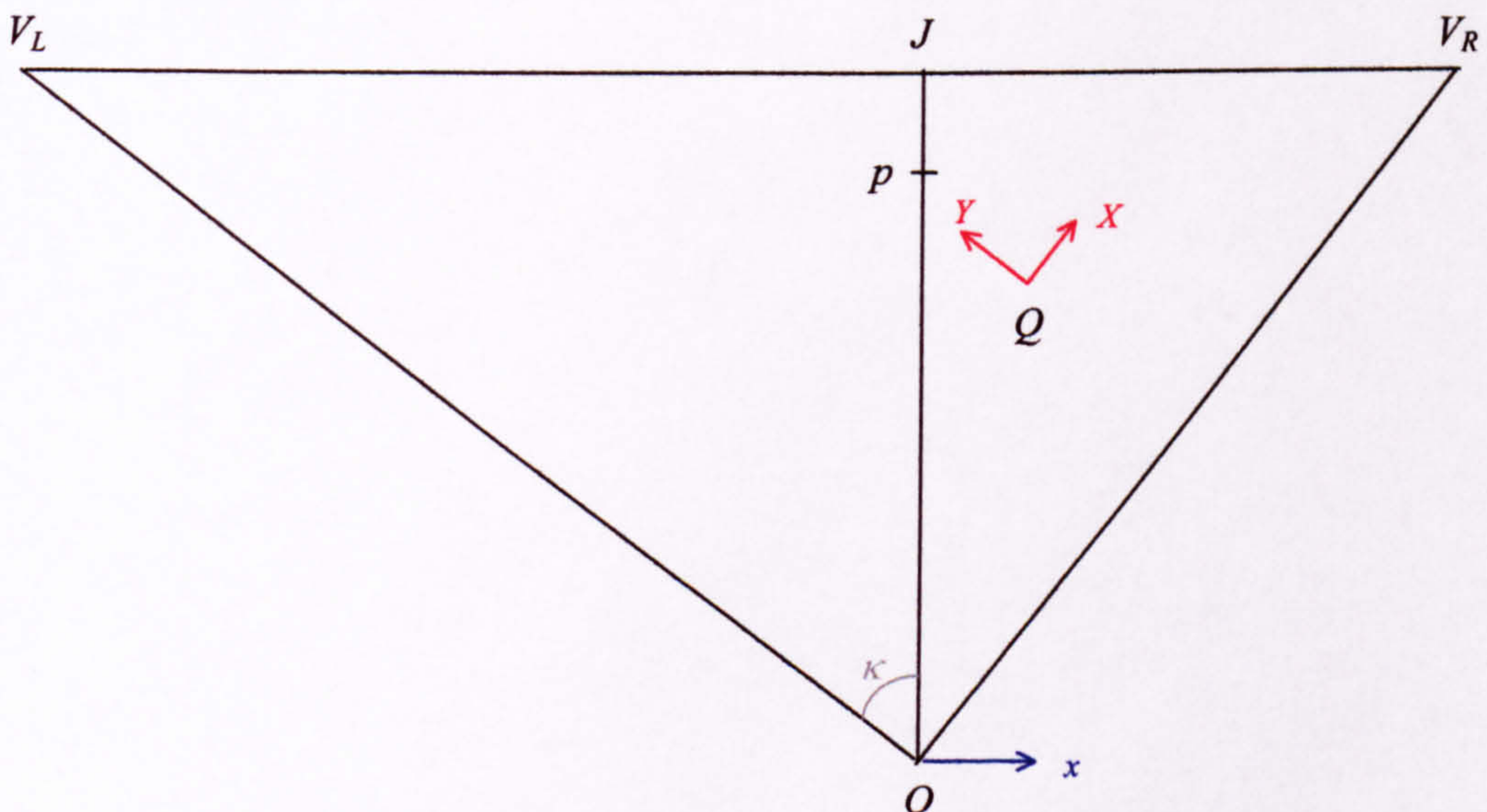


Figure 8.5: View of co-ordinate systems in elevation (a) and plan (b)

Figure 8.5. The second version of the Phoenix algorithms works by converting between two co-ordinate systems, those of the image and the object.

An intersection approach is used in the calculation of pJ , in case the two points V_R and V_L are not horizontal. If the difference were large, it would imply that the ICS had not been set up correctly, therefore this should be corrected before continuing. The value of pV_R can then be calculated by the use of Pythagoras to give

$$pV_R = \sqrt{(xV_R^2 + pJ^2)}$$

Equation 8.11

Using this value and similar triangles, V_R-p-J and V_R-K-V_L , it can be seen that the distance pK is given by;

$$pK = \left[(xV_R - xV_L) * \left(\frac{xV_R}{pV_R} \right) \right] - pV_R$$

Equation 8.12

This allows the distance of c to be calculated for the image using Equation 8.9.

The next step is to calculate the position of the camera in the OCS from the relationship between the co-ordinate systems, see Figure 8.5. This is worked out by firstly calculating the position of p in the OCS, and then calculating the position of O relative to p in the OCS. These two positions added together will then give the position of the camera, O , relative to the origin of the OCS, Q .

The angle from Op to OJ , γ , is given by

$$\gamma = \tan^{-1} \left(\frac{pJ}{c} \right)$$

Equation 8.13

and the angle from OJ to OV_L , κ , is given by

$$\kappa = \tan^{-1} \left[\frac{(x_{V_L} * \cos \gamma)}{c} \right]$$

Equation 8.14

Using these two angles, the separate parts of the position of p , in the OCS, can be calculated as

$$\begin{aligned} X_p &= -x_Q \cos \kappa + y_Q \sin \gamma \sin \kappa \\ Y_p &= -x_Q \sin \kappa - y_Q \sin \gamma \sin \kappa \\ Z_p &= -y_Q \cos \gamma \end{aligned}$$

Equation 8.15

Therefore the position of O is

$$\begin{aligned} X_O &= c \cos \gamma \sin \kappa + X_p \\ Y_O &= -c \cos \gamma \cos \kappa + Y_p \\ Z_O &= c \sin \gamma + Z_p \end{aligned}$$

Equation 8.16

With the position of the camera, O , known, all that is left to find out about the camera is the rotation ω . This can be seen from Figure 8.5 to be

$$\omega = \left(\frac{\pi}{2} \right) - \gamma$$

Equation 8.17

It is then possible to use Equation 8.7 to convert a point in the ICS to its related OCS point. The unknown in this equation, μ , is set by fixing one of the axes in the OCS to a value either directly or through the usage of a geometric constraint.

8.5 Summary

As stated earlier in this chapter, the second method benefits over the first by directly solving the solution rather than relying on iteration. With this, also the solution creates a final model where the Z-axis represents height as opposed to the earlier solution using the Y-axis.

Further improvement can be made to the Phoenix algorithm by the introduction of statistical analysis on the generated points to form a best-fit model as opposed to the rigorous solution currently created. In this way, more emphasis would be placed on the creation of correct geometric features rather than on the position of selected detail points. Enhancements to the method are discussed further in Chapter 9 based on the results of testing and use in a real situation.

9. Testing of Phoenix

To examine the functionality and accuracy of the Phoenix algorithms, various tests were carried out as part of the research. These were designed to test the relative accuracy and workload of the Phoenix approach compared to other, more traditional methods, as discussed in Chapter 6. The final test was to convert the final building model, as created by Phoenix, and a sample landscape into VR. This was to explore the techniques and related problems for the conversion of data to a suitable VR format.

At the time of the initial test, the Phoenix algorithms had only been developed to handle two-point perspective images. Further development was then carried out on the algorithms to allow the usage of three-point perspective images. The second test, detailed in Section 9.3, was then carried out using these algorithms. For the conversion to VR, an image of a historic building was used to research the potential differences in technique required for this type of image.

9.1 Test methods

In order to test the accuracy of the Phoenix approach, various methods were used to obtain the three-dimensional data. Including a mixture of traditional survey approaches and photogrammetric techniques, as discussed in Chapter 6, the methods used were

- metric photogrammetry
- theodolite angle intersection
- multiple oblique photographs using PhotoModeler
- single oblique photograph using Phoenix

Bottrill *et al.* (1998) compared these methods for ease of use, resource demands and accuracy of results. For the tests, a modern building (Martha's Bar) was chosen for which at least two sides of the building were accessible for survey. The decision to select a modern building was made to simplify the survey and allow the geometry to be easily constrained for all approaches. This avoided the fact that, for many old buildings, settling, and subsidence in the ground and building has led to the walls becoming bowed.



Figure 9.1: Modern day test building

Figure 9.1. For the initial testing of the Phoenix approach, a modern day building was selected to simplify the survey and the geometric constraints.

The initial stage was to set up a three-dimensional network comprising of four reference stations around the building. This was established using a Wild T1000 electronic theodolite and Wild D13000 distomat. From this, a local grid was set up with co-ordinates being assumed for one of the stations, and a bearing from this to another station being fixed. The information from the network observations of these points was processed in the network adjustment software, STAR*NET. This network formed the basic co-ordinate system for the four techniques tested.

9.1.1 Theodolite angle Intersection

The angle intersection survey consisted of vertical and horizontal angle measurements taken to a series of identifiable points on the building. This was done as a non-contact survey using features on the two façades of the building to form the points. Difficulty was found in accurately targeting the same point from different stations. This included problems of not being able to see the point, as well as accurate positioning on the point. For example, the far-left side of the building was obstructed from most of the stations by a drainpipe, as can be seen in Figure 9.1.

The distance measurements taken using the distomat were found too inaccurate to constrain the model significantly. This problem arose through the angle of incidence of the reading with the building being too large from some stations, along with the surface texture diminishing the signal. These readings were therefore removed from the calculation, and a pure angle approach was used. Through this, it was found better to fix the plan position before moving onto the full three-dimensional model. This removed the problem, where it was possible for the three-dimensional intersection rays from the stations not to intersect. Therefore, by restricting the two-dimensional position of the intersection before calculating the average height, a more accurate plan model is

produced. From this, a best height could be calculated for each co-ordinate by examining where the three-dimensional rays intersected the plan positions.

9.1.2 Metric photogrammetry

For the photographic methods, imagery was taken using a large format (65mm x 90mm) Wild P32 metric camera for the stereo photographs, and a Nikon 35mm SLR camera for the oblique photographs. The stereo photographic method required a baseline between the two camera positions to be made parallel to each façade being studied. This is to ensure that there is only one-point perspective for the building in the image. If this is the case then scale measurements can be read directly from the images with no rectification. The other factor that needs to be taken into account is the base to distance ratios for the photogrammetry. For the survey, acceptable ratios of approximately 1:4 and 1:8 were achieved for the front and side façades respectively.

Three-dimensional co-ordinates for the photogrammetric control points were added to the photography using information from the angle intersection survey. These control points were chosen to be points easily visible on each photograph for which an accurate position was known. The photography was then processed using a Zeiss Planicomp P3 analytical plotter and PCAP software, before the resulting model was exported in the DXF file format for use later in AutoCAD. Knowledge of the locations and the camera meant the model was created at the correct scale during the process.

9.1.3 PhotoModeler

The PhotoModeler software uses multiple images to generate a best-fit three-dimensional model of an object. Similar to stereo photogrammetric techniques, the PhotoModeler software follows a three-stage approach in gaining a solution for multiple

images. Firstly, the relative position of the camera for each photograph is found. This is done by use of a scaled cuboid being graphically manipulated until it matches size, shape, and depth of an object on the photograph. Once this is done for each photograph, the relative camera positions can be calculated. The next stage is to indicate corresponding points on the different photographs that construct the wireframe model of the object. To be fixed, these points must exist on at least two photographs, although redundancy can be added into the model by selecting the point on more images. This redundancy allows a tighter solution to the problem to be generated. The final third stage is an adjustment of all the points until a best-fit solution is generated.

As the final adjustment takes into account the position of all points marked on the images two models were made. The first contained only the points that could be accurately identified on all the images used, and the second used all points indicated to give a complete model of the building.

The functionality of PhotoModeler has since been extended to allow for the processing of single images along with multiple ones. Following a similar setup to the existing solution for multiple images, the user first fits a bounding box on the image around the area of interest. Camera parameters can also be added to the model if known to tighten the model.

9.1.4 Phoenix approach

For the Phoenix algorithms, the photograph had to be processed by hand, although the conversion from two-dimensional co-ordinates to three-dimensional positions was handled by a spreadsheet. In the later test, the processing was all carried out by the Phoenix software. The first stage was the setting up of the co-ordinate system of the image. As a calibrated camera was used for the photography, then the principal point of

the image was known and used as the origin of the image co-ordinate system. Lines parallel and vertical to the edge of the image through the principal point were then constructed for the x- and y-axis.

Next, identification of vanishing points on the image to the left and right of building was carried out. As the camera had been levelled before photography, it was known that there would be no V_D vanishing point for the image. The levelling of the image also meant that the horizon line would fall along the x-axis of the image. The vanishing points were then taken as the average point where vanishing lines crossed the horizon line on either side. Further tests to discover the effects on the model due to the choice of vanishing point position are discussed in Section 9.2.1. The user then enters the co-ordinates for each point of the model that they wish to calculate and a possible answer is given dependant on any constraints added. This spreadsheet used the first Phoenix approach, discussed in Section 8.3; therefore, OCS uses X and Z for the plan position and Y for the height. Once the object is completely modelled it can then be created inside a CAD package and scaled by the inclusion of a known dimension.

9.2 Test one results

The resulting models from the different tests were all input into AutoCAD for analysis, see Figure 9.2, and all appear visually similar to the control model, chosen to be that of metric photogrammetry. The photogrammetric survey was chosen as the most accurate form of survey, after the problems stated earlier with the theodolite intersection.

The images from the oblique photogrammetric techniques, PhotoModeler and Phoenix, were scaled using measurements taken at the site. All the models were rotated and aligned in AutoCAD using the corner as shown in Figure 9.2. In this way, the tests

would concentrate on the relative position of the points, rather than errors in positioning the models in the framework.

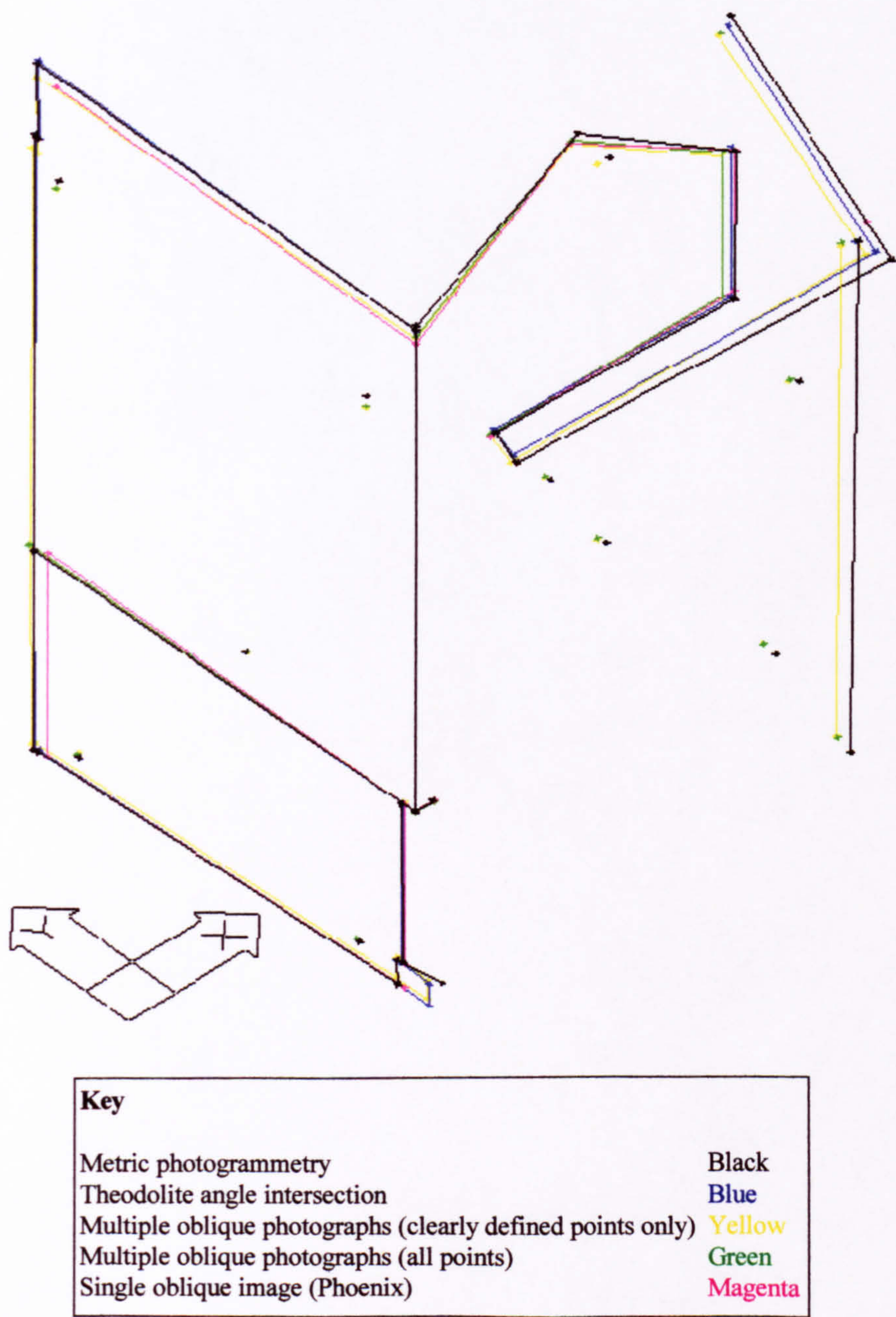


Figure 9.2: Comparisons of models from test one.

Figure 9.2. The models created by the different methods were imported into AutoCAD to be compared both visually and statistically.

For this research, the tests were designed to investigate the similarity between the models. For many applications, such as archaeology, an accurate representation is not

required. In this case, it is a ‘feel’ of the shape of the building rather than accurate measurements that aid the interpretation. Statistical tests were carried out on the position of each point in the models to establish the accuracy of the survey techniques. The models were tested against each other in pairs using the F-test for two population variances (Erricker 1976) to see if there was a significant difference between them. In this way, each model was compared with all other models created. The only models to pass this test, showing that significantly they are similar, were the two created by PhotoModeler. Bartlet’s test for equality of K variances (Kanji 1995) was then used to confirm this hypothesis, and showed that the models were all significantly different from each other.

Further statistical analysis was also carried out on the actual position of the points in the models. Table 9.1 shows a summary of the differences in X, Y and Z position of each of the models with that of the base model, metric photogrammetry.

Table 9.1: Summary of differences of models with base model

	Angle intersection			Phoenix Algorithm		
	δX (m)	δY (m)	δZ (m)	δX (m)	δY (m)	δZ (m)
Max. Dif.	0.174	0.216	0.107	0.376	0.408	0.346
Range	0.241	0.256	0.173	0.408	0.664	0.509
Mean	0.005	0.032	0.004	0.048	0.013	0.085
Stan. Dev.	0.051	0.061	0.039	0.116	0.201	0.152

	PhotoModeler - Accurate points			PhotoModeler - All points		
	δX (m)	δY (m)	δZ (m)	δX (m)	δY (m)	δZ (m)
Max. Dif.	0.218	0.281	0.203	0.282	0.337	0.238
Range	0.238	0.341	0.297	0.336	0.424	0.360
Mean	0.037	0.073	0.038	0.046	0.067	0.043
Stan. Dev.	0.057	0.100	0.078	0.074	0.102	0.097

Table 9.1. Comparison of test models with those produced by metric photogrammetry. The table shows the maximum difference and range of position along each axis, along with the mean and standard deviation of the differences.

The largest differences in position were seen in the Phoenix approach, 0.376m, 0.408m, and 0.346m respectively for the X-, Y-, and Z-axis. If Figure 9.2 is examined it can be seen that the greatest error occurs towards the extremities of the model, and the image. This implies that some of the error may be down to the accuracy of the selection of points on the image. Possible causes and solutions for this are further discussed in Section 9.2.1 and Section 9.4.

9.2.1 Location of vanishing points

The test included research into the effects of the location of the vanishing points on the model produced by the Phoenix algorithm. This was due to problems noted during the construction of the Phoenix model in the above test. For the results given above in Section 9.2, the average x-axis of all the positions for each vanishing point was taken; the y-axis was known to be zero as it was a levelled image. To test the outline of the model, it was recreated using combinations of the extreme positions for the two vanishing points of the object, see Figure 9.3. Through this, it would be possible to see how great a difference the selection of the vanishing point could make.

From Figure 9.3, a distortion in the model co-ordinates can be seen between the usage of the extreme left and right positions (the green and yellow lines). This implies the best approach is to use, as was done, the best estimate from all the points measured. This range of possible positions caused by the effects of the selection of points for each co-ordinate can explain some of the loss of accuracy of the Phoenix model. Section 9.4 explains further ways of enhancing the position of the vanishing point by use of a computer-based algorithm and by weighting sets of vanishing lines.

9.3 Police House

The second survey was carried out on a building on the Newcastle University campus. For this test, a direct comparison between the angle intersection approach and the Phoenix approach was used. This was decided as it was felt the angle intersection approach would be accurate for this model, as all edges could be seen clearly from the survey positions. For the photogrammetry, the test used a non-metric 35mm camera and tilted photography. This was to simulate, more readily, the real application of the Phoenix approach.

The angle intersection model was produced by a similar technique to that used for the model of Martha's in Section 9.1.1. Three stations were set up around the building, as shown in Figure 9.4, to enable two sides of the building to be studied. After the problems during the earlier test, where areas of the building could only be accurately surveyed from one station, the sightings from the proposed stations were tested before fixing their position.

Co-ordinates for station one and a bearing from station two to station one were fixed, allowing a local grid to be set up for the site. The control traverse was then surveyed

using a TC1010 between these stations. As before, the network observations were adjusted using a least squares analysis provided by the STAR*NET software. The positions were all fixed to within the 95% confidence level for a three-dimensional data set. This resulted in the full three-dimensional positions for each of the control stations relative to the local grid.

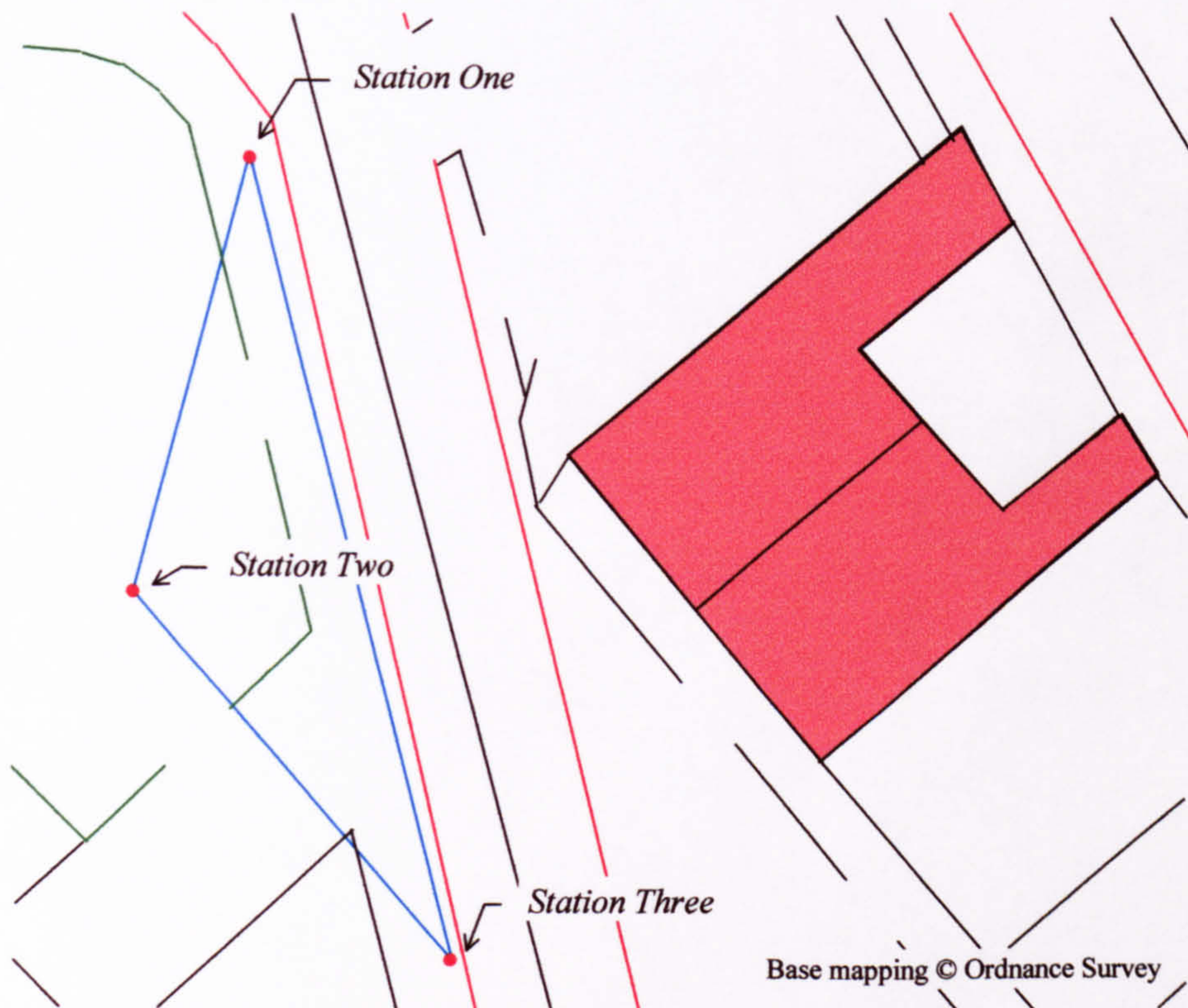


Figure 9.4: Location of Police House surveyed

Figure 9.4. The traverse for the second set of test data was carried out with the problems of the first test in mind.

Once the traverse had been calculated and fixed, observations were made to points on the building from each of the stations (see Figure A10.1 in the Appendices section). This required the observation of three rounds of vertical and horizontal angles to each of the required observation points, ensuring every point could be seen from at least two stations. The three-dimensional co-ordinates of points on the façade of the building were then added to the problem in the STAR*NET program and a final least squares

solution produced to give their position (see Appendix 1 for this data). As with the initial traverse, this information was calculated to within the 95% confidence interval. Full results from the survey as produced by STAR*NET can be found in Appendix 1 and Figure A10.1. As before, errors in position occurred due to the difficulty of targeting the same point from different stations. This problem can be decreased with the aid of other approaches, or equipment for example, by applying targets to the object requiring measurement. Alternatively, two theodolites with laser dot targeting can be used to ensure exact positioning. For this research, neither of these two methods were available to the researcher.

The second part of the test was an investigation of the use of a single image to produce a wire-frame model. The Phoenix algorithms had been improved and incorporated into a Windows application since the test on Martha's bar detailed in Section 9.1. The version of the algorithms used in this test is as explained in Section 8.4, allowing tilted imagery to be used.



Figure 9.5: Single image used for model of Police House

Figure 9.5. Example of image used for the second test of the Phoenix algorithms. From the sloping of the vertical walls, it can be seen that the image is tilted.

Therefore, for this test a tilted image, pictured in Figure 9.5, displaying three-point perspective was used. As stated, the photograph was acquired using a non-metric 35mm camera, then scanned at a resolution of 300dpi before processing with Phoenix. For the test, the second version of the algorithms, detailed in Section 8.4, were used, with the software written for the research. With all this information, the relative position, O , orientation, ω and κ , and focal length, c , of the camera can be obtained. Once this was set up, points along the X- and Y- axis of the model were then selected on the image for which three-dimensional co-ordinates were calculated.

9.3.1 Results of second test

The model produced by Phoenix was then translated, scaled, and rotated to match the model produced by the angle intersection survey. The small sign found on the left face of the building was used as a reference for the transformations. Figure 9.6 shows a comparison of the two models when they are overlaid.

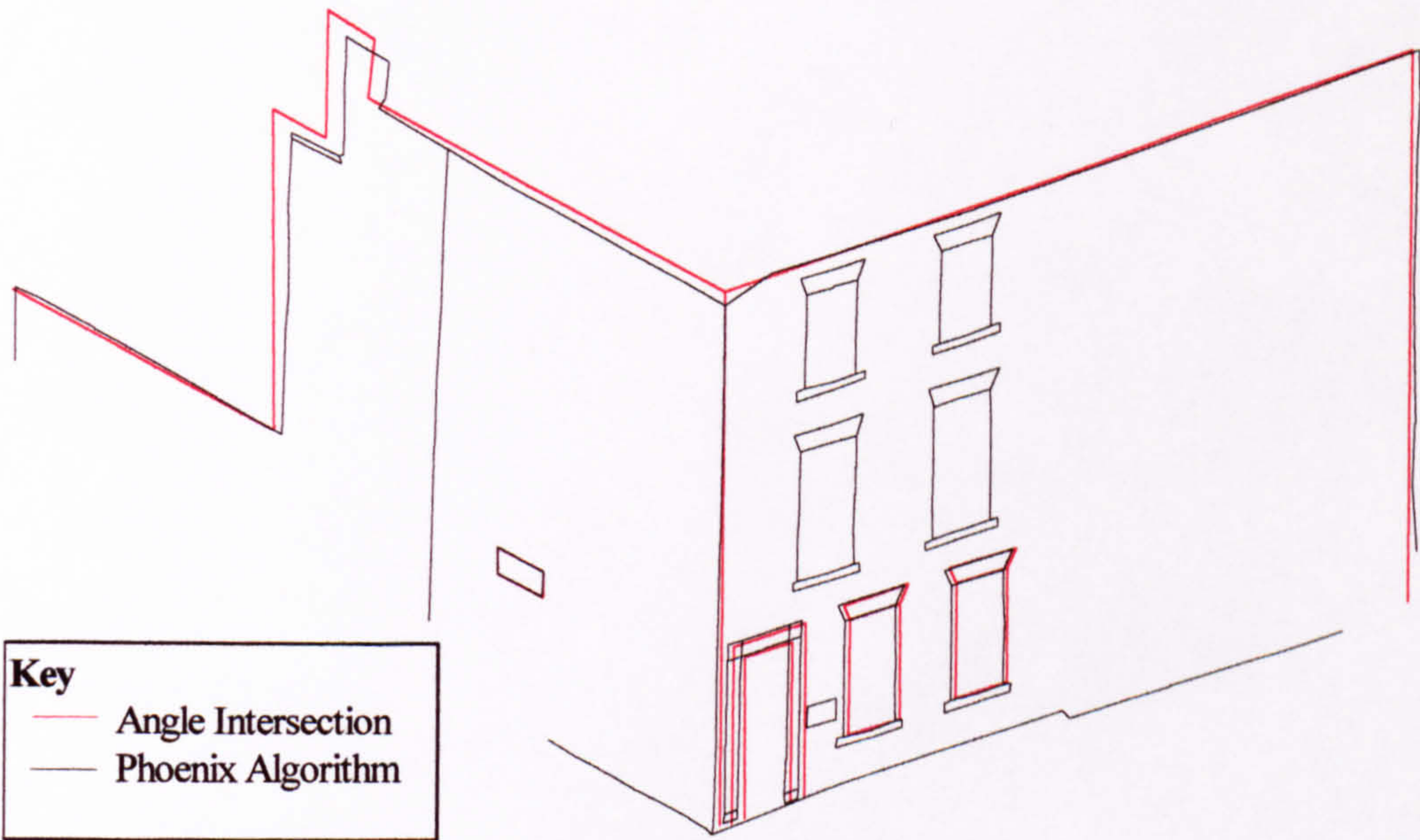


Figure 9.6: Comparisons of Phoenix and surveyed models

Figure 9.6. When overlaid it is possible to see the differences between the two test models. As can be seen there is a greater error in the left-hand side of the model.

Visually, it appears that the models are similar although a greater error is apparent towards the back left-hand side of the models. As well as a visual comparison, the models produced by the two methods were compared statistically. The difference in X, Y, Z co-ordinate, and position of each point were calculated and the difference in distance between pairs of points was calculated. The full data for these tests appear in the appendices in Table A1, Table A2, and Table A3. Only points and distances appearing in both models were included in the tests. Table 9.2 shows a summary of the main results found. The δ Dist results check the distance between every pair of points in each model. The error percentage is the percentage difference between the distance for the angle intersection model and the Phoenix model.

Table 9.2: Results of statistical tests on models from test two

	δ X (m)	δ Y (m)	δ Z (m)	δ Pos (m)	δ Dist (m)	Error (%)
Abs. Max	0.541	0.322	0.201	0.650	0.780	30.7
Abs. Mean	0.133	0.104	0.069	0.190	0.183	2.4
Abs. SD	0.140	0.080	0.055	0.161	0.200	2.7

Table 9.2. The results of the statistical tests carried out on the second test of Phoenix compare well to those from the first one.

These results show that the average error, 0.190m, in the position of a point is approximately the same as the average error in a distance, 0.183m. This difference in distances can be represented as, on average, a 2.4% difference between the two distances measured. The full tables of these results can be found in the Appendix section of this research.

Examination of the full table of results shows it is possible to see that if the outline of the left side of the building is ignored, and only the front outline and detail examined, then the average error falls. This produces a change from 0.190m to 0.123m in position and 0.183m to 0.063m in distance. The average percentage error in distance drops only

slightly to 2.0%. This implies that the result of an average error in distance of around 2.0% is what can be expected from the technique at the current stage. The following Section discusses the possible causes of errors in the technique, and suggests how it may be improved.

9.4 Discussion of errors in Phoenix approach

Errors in the calculation of a position by Phoenix can occur in many stages throughout the process. The main error factor is the accuracy with which a user can select a point on the image. The image used in the second test was from a print which was 8" x 6", or 202mm x 152mm, when scanned at 300dpi this produced a bitmap of 2375 x 1800 pixels. The resolution of the scan can be increased, although further difficulties can then be produced due to the size of the file. A bitmap of the above dimensions will take approximately 3 MB to store. By increasing the scanning resolution, the file size soon becomes unwieldy.

As each selection of a point on the screen by the user is able to produce error then these errors can accumulate through the process, from the initial selection of vanishing points onwards. To be able to track and account for this error it would be necessary to apply a weight to each point that is selected. This error value would be mainly based on the resolution at which the user selects the point, although other factors should be considered. For example, when producing vanishing lines to use later to create vanishing points, longer lines are liable to be more accurate and should be weighted accordingly. Presently in the Phoenix software, equal weighting is given to each intersection. A refined approach would be to base the weighting on the relative length of the lines intersecting, to those in the complete set selected. In this way, two long lines will influence the final point more than two short lines, or a short and a long line.

Another source of error now in the Phoenix software is that it is assumed there is no rotation of the image plane around the z-axis. Where the vanishing points are not on a horizontal line, suggesting a rotation, the intersection with the y-axis is used. A better approach would be to include a tolerance on the angle about z and allow the image axis and the object axis to be rotated about all three axes.

The greatest distortion between the two models was found at the points furthest away from the origin of the ICS. Possible reasons for this increased error factor can be seen. Firstly, distortion in the lens is liable to be greater near the extremities of a full frame image, and secondly, the effects of perspective are greater at the edge of the image. With perspective effects, for example, the vertical edge of the building nearest to the image plane is 119mm, or 1410 pixels, long on the image. This represents a distance of 9.169m on the object as calculated from the theodolite angle intersection. A similar object distance measured along the length of the drainpipe to the right on the image is only 95mm or 1122 pixels. Therefore, the real world distance represented by a pixel has increased from 6.5mm to 8.2mm. Although this appears a small increase, there is also the possibility that a user may not exactly select the pixel they require. Tests to see how well a user can select an actual pixel have not been carried out, but would be required.

9.5 Historic Model

The final test case involved the conversion of an historic image to a three-dimensional model. This test was designed to examine the suitability of the algorithm and software to processing a historic image, as well as to further examine the problems of moving spatial data into a VR world. The processing of historic imagery includes additional difficulties, for example, the curvature of walls, degradation in the clarity of the image,

and the possibility of a cropped image. Figure 9.7 shows the image used for the test, representative of the other historic images available to the research. The building processed for the test is the one located in the foreground of the picture.



Figure 9.7: Historic image processed by Phoenix

Figure 9.7. Image processed by Phoenix to test the capabilities of processing historic imagery and converting the resulting model to a VR world.

For processing by the Phoenix software, the image was scanned into the computer at 300 dpi. The origin of the object was taken to be the nearest, bottom corner of the overhanging part of the building. This point was chosen due to the clarity of the image; the other prospective corners were hidden in shadow. The problem of curvature in walls can be seen in this image, especially on examining the line of the pitch of the roof. This makes it difficult for the current Phoenix algorithms to model accurately the roofline, and for this test, it was assumed straight. Further difficulties arose from not being certain that the walls themselves were perpendicular to the surface. The model of

the building was then transferred to AutoCAD for final processing and the creation of a faceted representation for the VR software.

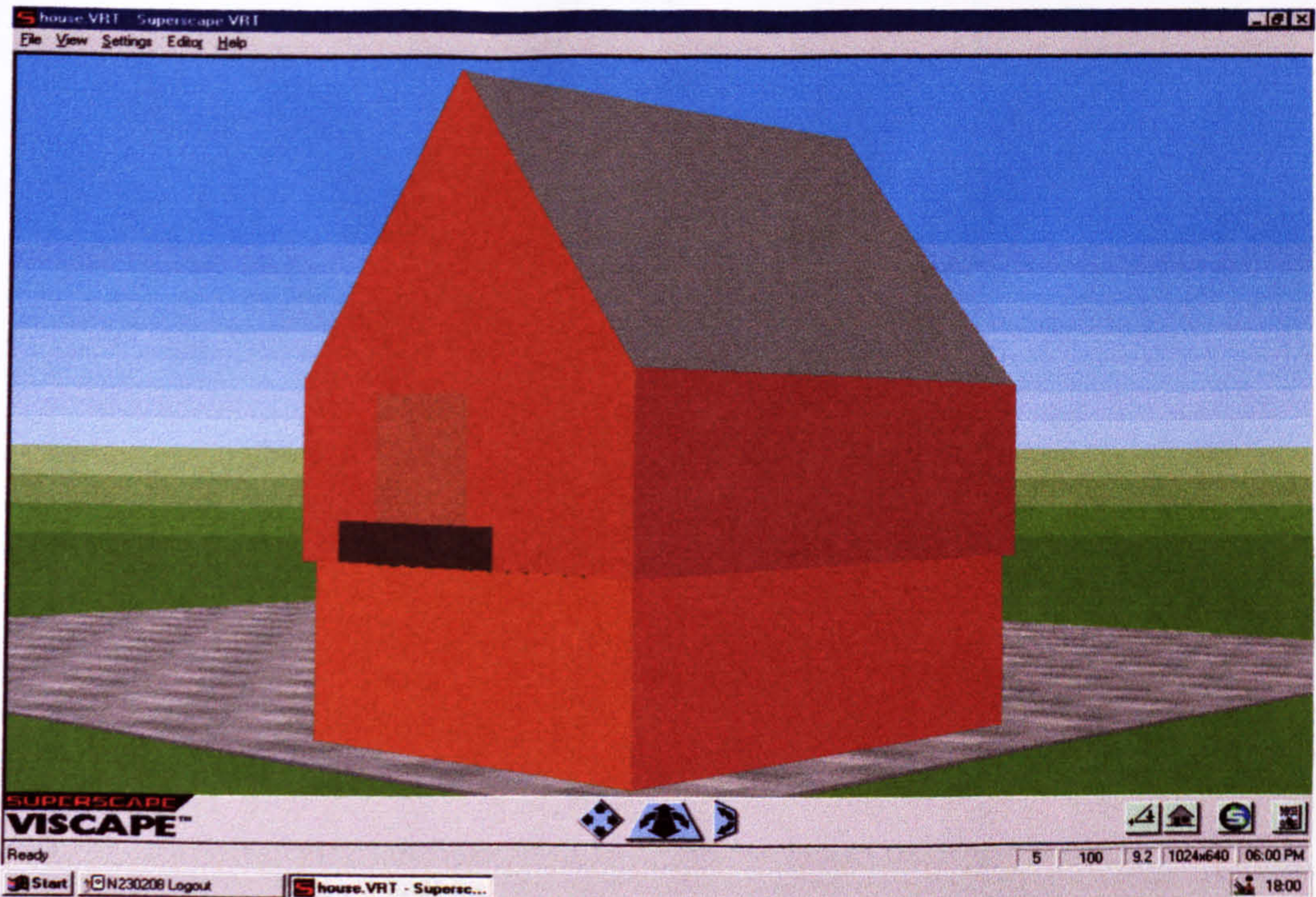


Figure 9.8: VR model created from historic image.

Figure 9.8. The VR model created from using the Phoenix software on the earlier historic image.

Figure 9.8 shows the final model as displayed in the VR software, Viscap. As can be seen the approach has produced a visually similar model to that in the image, although the same statistical differences as before will apply. This three-dimensional model can be scaled by approximating heights for standard objects such as the doorframe or the windows. Scaling can also be applied to the model by using neighbouring buildings and features for which dimensions may have already have been calculated, or that may still exist. This allows measurements to be taken to gain a greater understanding of the architecture of the time.

9.6 Summary

The tests show that although the Phoenix approach produces a visually similar model, the current algorithm has significant differences, approximately 2% difference in distance, to more traditional methods of surveying. However, as suggested earlier in the chapter, for some applications all that is required is a visual model. Further analysis of the model during construction would allow for a more accurate final model.

The research suggests that a combination approach is the best method for surveying buildings in three dimensions. The initial framework of the building is best constructed using an accurate three-dimensional survey of the building by theodolite. The detail of the façades of the building can then be added by photogrammetric techniques, using either stereo or rectified photographs.

What the Phoenix approach does supply is a fast and efficient method for the construction of a three-dimensional model that Debevec *et al.* (1996) state is required. In addition, as shown in the final test, the Phoenix approach allows a model to be created from a single oblique image, a process not possible using photogrammetric approaches. This allows for the creation of historic townscapes to be constructed for use in research work by archaeologists.

10. Conclusions

The main objective of this research was to examine, and develop methodologies for, the creation and distribution of higher dimensional geographic information. To illustrate the ideas and the technologies discussed, the concept of creating a representation of virtual historic newcastle was used.

By exploring the existing and future technologies required for creating such an application, the main aims, as stated in Chapter 1, were to

- Review current methods and technologies for displaying geographic data on the Internet.
- Discuss the structure of three-dimensional data for use in virtual reality and Internet applications.
- Examine the requirements and techniques for creating data sets that contain three and higher dimensions.
- Create a new methodology to extract three-dimensional information from single, oblique, images.

The research investigates how the Internet, distributed computing, and techniques for generating three-dimensional models from existing and new data could enable the creation of VHN. The research examines a broader spectrum of ideas than just those involved in the standard areas of current research into advancements for GIS, encompassing; machine vision and computer gaming, along with various data capture and structure techniques. The final stages of the research dealt with the particular case of extracting three-dimensional information from a single oblique image of an object, and developed new methodologies for this task. This part of the work constitutes a significant contribution to three-dimensional data capture techniques. This final chapter

highlights the conclusions arrived at during the research to draw together related statements and results given throughout the thesis.

10.1 Display of Geographic Information

The implementation of VHN requires the ability to display geographic information in a three-dimensional format, so the user is free to explore and experience the environment. The research raises the comment that today the term ‘geographic information’ can encompass a wide range of data sources and types. The main software applications in which geographic information is currently used are GIS and CAD. Over the period of this research, there has been an increase in usage of spatial data for other applications, for example, spreadsheets, word processors, and the web.

Computing power has reached a level that allows geographic information to form an interface into other data sets and worlds. This enables the multi-dimensional GIS discussed in Section 2.1 to be realised. Unfortunately, many existing applications restrict geographic information problems to two dimensions, constrained by the data and methodologies used.

The concept of digital geographic information is now reaching a broader cross-section of the computer population (Section 2.3). Traditionally regarded as the vector and raster cartographic data used to produce mapping, geographic information has since expanded due to the increased usage of computing. More sources of information may now be said to have a geographical footprint, for example, books, photographs, sounds, and videos (Kemp 1995). The amalgamation of these sources can give a more defined model of a geographic location, especially in a historical context, than mapping alone. For the VHN application, these data sources include historic photographs, paintings and

documents that have a geographic location, defined by the place they describe rather than an exact co-ordinate system. The VHN project can add value to these by allowing the researcher to immerse themselves in the environment created, for example, by allowing them to see the painting in the context of the position it was painted from.

Taylor (1991) suggests there are three components of visualisation, stated as cognition, formalism, and communication. These can be represented as a triangle (see Section 2.1). The research suggests that this triangle does not accurately represent the current situation. The illustration of the application of these technologies by Collinson (1997) (see Section 2.4, in particular Figure 2.8) suggests that Taylor's triangle has become disjoint. Cognition has been left in a two-dimensional arena while the other two sides of the triangle have moved forward to the third dimension. Developments in formalism mean that the communication of geographical information is now possible in three dimensions. This raises the question as to why GIS applications have not moved to encompass all the available technology.

Increasing computing power is being fully utilised in the three-dimensional environments created for the latest games. Coupled with this, the games industry has now begun to use accurate large-scale spatial data for simulation and sporting events, as discussed in Section 2.2. This allows an insight into the use of spatial data for displaying information, but as this research shows in Section 2.1, it is not the view seen in GIS applications.

The research and development of the VHN application suggest that there are two main barriers to the full utilisation of three-dimensional data. Firstly, the effort required in moving existing data from two-dimensional data sets to three-dimensional ones, and secondly the lack of definitions for how this data should be structured in three

dimensions. The next two sections of this chapter detail the conclusions from the research on these two points.

Technologically, GIS research is standing on the shoulders of the other sciences, such as CAD and the Internet, and picking off ideas from them. This technique has worked for GIS in the past, although in some areas the concepts cannot easily convert from aspatial to spatial information. Without employing current ideas, GIS will become left further behind, while lacking experience to build on later. Already the level of three-dimensional visualisation and interaction in CAD and computer games exceeds that of the high-end GIS applications. Through collaboration between GIS developers and other technologies, this ground can be made up.

In many cases, the GIS products available today were born out of CAD products or CAD companies, for example Autodesk, the creator of one of only two transfer formats used for the supply of OS digital vector products. With this change of direction in the industry from CAD to GIS, the co-operation between the CAD and GIS disciplines has declined in many cases. Early CAD systems were basic two-dimensional electronic drafting tools, functionally equivalent to the tools available for hand drawing. The increased use and influence of these tools on the manufacturing industry saw the requirement for three-dimensional modelling, first in wireframe and now with solid modelling. This gave birth to the current range of CAD, Computer Aided Manufacturing (CAM), and Computer Aided Engineering (CAE) applications. As CAD has moved into three-dimensional data, modelling, and analysis, GIS has remained with a two-dimensional outlook on the world. Although GIS is a much younger industry than the underlying CAD technology, the algorithms for the display and manipulation of spatial vector data are the same. Basic GIS extends CAD functionality by adding

database access and statistical testing and analysis to the CAD data model. Instead of upgrading this data model in conjunction with CAD, current development has restricted the geographic information data model to two dimensions. Even where the same company develops CAD and GIS products, as with Autodesk, the availability of three-dimensional functionality exists in the core product but little use is made of it in the GIS application.

The research suggests that one solution to inertia in GIS to embrace full three dimensionality is in the use of the evolving component and Internet technologies that are currently having an influence on the software market in general. The next section details the capabilities of these, and how they can be utilised by projects, such as VHN, to allow for the distribution of three-dimensional geographic information.

10.2 The Internet

As shown in the sections above and Chapter 4, the Internet is having an increased influence on the design of software. With this influence has come the evolution of text based file formats, for example, HTML, VRML and XML. Although applications (such as word processors, spreadsheets, and databases) have developed to interpret and produce these, GIS, with its graphical nature, has not moved so fast. As shown in Sections 4.4 and 4.6, representations of two- and three-dimensional spatial information can be stored in VRML and XML. The GIS community has done little to influence the development of these standards, and currently these formats do not perform well with spatial data, due to its inherent irregularity. VRML was developed for basic VR models that are more geometric, and SVG, a graphical implementation of XML, has roots in desktop publishing. The use of XML as a database transfer format, for example in Oracle, could see an increased use in GIS as a neutral format. Extending XML to

enable the storage of graphical data, using SVG, creates a simple format for GIS that has the capability to extend further. For this, a standard geographic document type declaration is required for the XML tags used to markup the data; a starting point is the development of spatial operator standards.

The VHN project, as illustrated in Figure 3.1, uses various aspects of HTML to provide the user interface to the underlying database of information. The two-dimensional image map allows for a visual representation of the user's current position in the three-dimensional world. This map also supplies a means for moving around in the three-dimensional model by selecting locations on the two-dimensional map. The text at the side of the virtual world is kept up-to-date using Dynamic HTML to alter the display as the user moves around the world. Using the document object model of HTML, the various software components on the page are integrated in the browser. This supplies a user interface to the underlying components for interrogating the textual and graphical data stores for information. With components, these data stores can be as easily based on simple text files as they can on high-end databases, for example Oracle.

Thirty years ago the network of cables and routers that form the Internet were just being put in place, and it is only seven years since the Internet was opened to all, as discussed in Section 4.1. Since then, the growth in the number of networked computers has meant that most businesses are connected, and the usage of email and web addresses is commonplace. For example, Microsoft now considers that all new computers connect to the Internet, allowing Windows 98 users to only update online. With the development of Internet2 (UCAID 2001), allowing faster data transfer, access through Digital TV, and game consoles with integrated modems, such as the Sega Dreamcast,

the growth in Internet usage will continue. This includes the usage in Internet aware applications as well as Internet based data sets.

Awareness of Internet technology is also growing in the eyes of the public, users, and developers. This is demonstrated by recent Intel advertisements for the Pentium III chip, which suggested it would enable users to work faster on the Internet. Many applications, for example, word processors, spreadsheets, CAD and databases, now include abilities to save files in a web format, or to access and utilise files stored on networked computers. It may well be that this move from the industry as a whole towards distributed systems and three-dimensional display may give GIS the push it is missing from the users who are content with a two-dimensional environment.

10.3 Component technology

The move to distributed information and applications is one that the whole of computing is currently examining. Once data and applications can remain at their original source, fewer problems arise from differing versions existing at multiple locations. Updates and maintenance of related documentation (technical and metadata) can then also occur at a central location. For data, an electronic record can be maintained of user access and editing, giving a complete history of the background. The next stage is to move to object-oriented data that encapsulates the metadata, before finally encapsulating related processes and functionality into the data, allowing interoperability.

Adding interoperability to data allows descriptions of itself and its processes to be accessible by any system, from any computer. Current computer usage is application centric; leaving it up to the user to decide which application is best for the purpose of

their work. This research has shown (Chapter 3) that future GIS, and computer usage in general, is liable to change to a data centric view of the world. Here, the system will decide which application or object is most suited to carry out a task on a piece of data. These systems will use objects available locally on the host computer, though possibly those available on a computer anywhere else in the world. As discussed in Sections 3.2 and 3.3, this depends on the agreement of standard interfaces between data and software components. These interfaces are not only to allow for the data and systems to communicate with each other, but also for efficient querying of large data sets. Current developments to SQL, Section 3.4, and other relational database systems include extensions to support multimedia and graphical data. As suggested above, development is designed towards Internet technologies but there is no restriction to applying it to geographical data.

By employing the ideas of object-orientation in both applications and data stores, GIS can take advantage of new ideas in areas such as visualisation, while remaining with the legacy of two-dimensional systems as well. The requirement to upgrade complete systems to add new functionality hinders the progression of GIS, moving to an open system will allow new software to build on existing. The concept of plug-and-play has been realised in hardware, but now needs to encompass software as well.

One of the benefits of the component systems discussed in Section 3.3 is that all areas of computing can benefit from advances in others. With this, GIS applications could access the three-dimensional graphics kernels available for the display of data in CAD. In a similar approach to hardware becoming components to allow for the easier extension and upgrade of systems, software can now develop in the same way.

The aim behind applying a component system model is to allow modules of GIS software to work and share information in native formats. In this way, the capabilities of all areas of GIS improve, removing the requirement on users to obtain their complete system from one manufacturer. As shown in Chapter 3, this is not currently possible due to the lack of industry standards to define objects, although improvements are being made. In the current computer industry, there is always something better in development than what is currently available. The ideas behind component technology should aid by allowing faster development and redesign of software to encompass those areas that have changed. However, until the majority of applications begin to employ object-oriented techniques, the situation will remain where it is today. During this research, various component systems and standards have developed, for example; CORBA, OGIS, and COM, as discussed in Chapter 4. As of yet few commercial GIS applications use the technology and ideas available to them from this.

Current GIS systems do not fully allow for the creation of three-dimensional distributed environments, or for the use of the range of geographic data types stated earlier. While some systems contain sections of the required functionality, no one system can do everything. Therefore, this research looked at using existing component-based technology to facilitate the creation of a GIS (Chapter 4). This allows the system to be upgraded and added to later, by the inclusion of new components rather than the overhaul of a complete system. The use of components allows for the development of a system now that can utilise three-dimensional data with the ability to be 'upgraded' in the future to take advantage of the latest graphics capabilities.

10.4 Moving from two to three dimensions

The research suggests that the delays in moving to three dimensions come from users finding problems with manipulating three-dimensional information in their minds. Possibly, this is through the lack of a metaphor for display, such as the map for two-dimensional data. Collinson (1997) suggested that for imaging three-dimensional information, it is necessary to present a true representation of the world, as illustrated in Section 2.4. This sort of representation is not possible with the current available component systems and is reserved for high-end machines. For the VHN project there was a requirement for the application to be available on a basic machine.

The conceptual difference between two- and three-dimensional information initially causes problems. Formal concepts for the generalisation and symbolisation in a two-dimensional view are defined (Keates 1989), the difficulty is in applying and extending these to a three-dimensional display. This is caused by the common belief that three-dimensional data must copy the real world for it to be an enhancement over existing two-dimensional data set and display (Collinson 1997). This implies the only advantage of the third dimension is for 'pretty' displays.

Based on McLaren and Kennie's (1998) classification of terrain visualisation techniques, Figure 2.7 suggests virtual worlds have a high symbolic content and closely match reality. However, it is important for the user to be able to separate the dimensions of a three-dimensional symbol from the dimension and position of the real world object that it represents, as is currently done for a two-dimensional map. Much of the current research into three-dimensional environments, for example in the games and entertainment arena, is based on making three dimensions realistic. Instead of two-dimensional cartographic maps, it is possible to have fully immersive three-dimensional

worlds to explore and analyse. Geographic information is no longer just a visual display of the world as it was with cartography. Currently, the main use for the three-dimensional visualisation aspect of a GIS is display. GIS needs to develop, so that visualisation becomes part of the whole process of GIS analysis and design, instead of just a product. Systems such as the ArcView 3D Analyst, detailed in Section 2.1, have moved forward, allowing the user to select and query objects from a three-dimensional interface. The range of functionality applicable to a three-dimensional object is still a minor subset compared to that for two-dimensional objects. The topological information stored with the three-dimensional model needs to be increased to allow a greater range of analysis.

What GIS requires from virtual worlds is the ability to display large quantities of data and the ability to employ symbols, as with current two-dimensional mapping, to display some of the complexity of the model. This research suggests that GIS would be following down the wrong route in aspiring to realistic data sets and should stay with a level of detail that allows the user to understand the space of the data.

The research has shown that the conversion of geographic information can only happen if both the information itself and the applications are developed. The current lack of three-dimensional data for spatial information is an inhibition in the uptake of applications that would require such data. GIS and geographic information are too inter-related to develop independently in different periods. Many developers argue that GIS does not need to be three-dimensional, as the data is only two-dimensional. In reverse, most data capture is two-dimensional, as most systems cannot completely handle three-dimensional data. As stated in Chapter 2, currently only the input and output of geographic information is three-dimensional, with basic object selection in

some systems. Most storage and analysis occurs in a two-dimensional world, with height converted into an attribute of position. True three-dimensional data sets are now becoming available through data techniques such as airborne laser scanning; yet, a defined range of applications beyond visualisation for this data is still not there.

The lack of three-dimensional applications for spatial information might rest on the data available for these systems (Chapter 6). Current spatial data forms a two-dimensional interpretation of a three-dimensional space. Chapter 6 describes the range of techniques that are currently available for the capture of three-dimensional information. Not all of these, for example the structured light methods in Section 6.6, are immediately applicable to large spatial features. These methods are included to show the range of approaches that can be used to capture the three-dimensional size and shape of an object. The basis of the Phoenix algorithms, Chapter 8, came from research into artificial intelligence to allow basic machine vision. Extending the approaches to three-dimensional data capture used by other disciplines will allow new techniques to be developed for the capture of spatial information on a large-scale. For example, laser altimetry or LIDAR, discussed in Section 6.3 is a direct extension of the smaller laser scanning approaches for capturing the shape of small objects. One restriction with the current methods of capture of three-dimensional data is the increased time, and hence cost, of survey. The true value of these new data sets can be seen when they are combined with existing data from land surveys for interpretation. By combining this new data with existing data, fast-automated methods can be used to generate basic three-dimensional models, which can later be enhanced.

10.5 Structuring data for the third dimension

Presently, large-scale spatial data mainly uses a vector data structure for the storage of information. Raster data, when it is accessible, is used for backdrops to the vector data or as a two-and-a-half-dimensional terrain. Full three-dimensional display using voxel technology, as described in Section 5.2.1 for the games and medical industry could now be applied for geographical display. Originally, data storage and processor speed has excluded the usage of large voxel data sets for display of geographic information. However, with processor speed and disk storage increasing, along with dedicated three-dimensional graphics cards, the processing of large models is possible. Enhanced spatial algorithms are possible through the voxel data structure, for example those calculating cross-sections and volumes. The computing power now exists to consider the storage of data in a vector format for space, and the direct conversion to a raster, voxel, structure when required for display or functionality. The restriction to this is the relative ease of conversion from vector to raster compared to the reverse approach. Currently, no GIS software converts data on the fly dependent on the required application of the data. Although a vector to raster conversion occurs for display, the data is stored in the operating system, and not utilised by current software.

The move from two to three dimensions requires not only a change in the use of data but also a change in the capture of data. The structuring of data for the use in a three-dimensional environment does not just involve the addition of a height co-ordinate to the existing planar ones.

An influence on the structure of geographic information is the data capture process used to obtain it. In the main, present data capture techniques aim to convert the three-dimensional world into a two-dimensional plan. It is necessary to change this

conversion from the data capture stage, to the data display stage. This allows for full three-dimensional storage of data, if not display. Until three-dimensional data is available, the concept of three-dimensional GIS will not become a reality.

Due to the history of GIS, as discussed in Chapter Two, the majority of GIS data formats tend to be two-dimensional, a restriction enforced by the software that uses them. If geographic information is to move to three dimensions, then these structures need to be able to extend. Along with discussing existing GIS structures, Chapters 3, and 5 proposed new formats in which to store geographic data.

With increasing usage of networked data sets, storage size could again become a factor in the structure of data, as discussed in Chapter 5. Currently, only those with direct links to a network, for example universities, can fully experience the Internet. To many, real-time multimedia and audio over the Internet are still not available. This means there is no real emphasis from users to develop this capability for applications. The current limitation of access speeds is often from the modems used to create the connection to the Internet Service Provider (ISP). The development of the Internet2 standard, being tested by leading American universities, will see increased data transfer speeds. Faster access times will also come from developing modem technology, and the implementation of modem access via cable and digital networks. With this, the true possibility of distributed GIS will be available.

10.6 Gathering Information from Single Images

The generation of accurate information from a single (oblique) image allows for faster data capture.

“There is a call for a method to conveniently acquire photo-realistic models of existing architecture” (Debevec et al. 1996, page 11).

Today, much of the research in the GIS arena is also being investigated, or solved, by other scientific disciplines. The initial basis for the Phoenix algorithms for single imagery, detailed in Chapter 8, was from photogrammetry. A wider foundation for the approach and solution came from examining research from the artificial intelligence community into machine vision. More co-operation between scientific disciplines is required to enhance and further the area of spatial science. Historical techniques such as the graphical ones discussed here, initially rejected due to workload, can now become possible as there is the computing power to perform the calculations. Many of the concepts for the single image approach came from techniques originally created for construction by hand drawing. With computerisation, gaining three-dimensional information from an oblique single image has become possible in a fraction of the time it takes to perform the task by hand.

Part of the problem with data capture remaining two-dimensional is the time taken to capture an object in three dimensions. This increase occurs through the amount of extra detail deemed necessary for capture in three dimensions, for example doors and windows. This research developed a solution to this with the Phoenix approach to extracting information from a single image, detailed in Chapter 8, allowing the creation of a three-dimensional model in hours rather than days.

Current photogrammetric techniques for single images are restricted to the use of orthogonal images. By allowing the use of oblique photography, the selection of images that can be used is increased. For the particular case of this research where historic images formed the main basis, this was a necessity. As Chapter 7 illustrates,

various methods currently exist in other disciplines for the extraction of three-dimensional data from a single image. This forms another example of how GIS research can benefit from examining the work of other disciplines. The artificial intelligence and the military communities have carried out the major research into the topic of single image rectification. These communities have concentrated more on the automated extraction of information from an image to enable the processing of large quantities of data and machine vision.

This research followed a combined approach, merging knowledge of the object from the user, with that gained from the mathematics of the perspective image. During the research, development of the technique was to illustrate a concept, rather than to produce a model in complex cases. Recent developments in commercial software (Section 8.1.1) have shown that this concept has a commercial requirement. However, here the emphasis has been on reconstruction for web pages and governmental usage rather than for the capture of spatial information.

As shown in the results of the tests (Chapter 9), the model that was produced by the current Phoenix approach is statistically different to that constructed by more traditional survey techniques. There is a 2% mean difference between a distance measured on the model and the equivalent real-world distance. This may be considered high for large-scale three-dimensional modelling e.g. internal building detail, but is acceptable for many applications. The immediate use of the technique was to allow for the three-dimensional visualisation of historic buildings and monuments. In this case, photography is often the only remaining data on the size and shape of the object.

In Chapter nine the techniques for three-dimensional data capture provided by the Phoenix algorithm, were shown to work with sufficient resolution for the intended purpose, i.e. population of VHN (Taylor *et al.* 2000).

10.7 Future Work

Future work has to be carried out to investigate the rich uses for three-dimensional data. In the main, only basic ideas exist, such as intervisibility analysis, this is because three-dimensional data has not been utilised enough to suggest any better function than visual impact. Methods for the rapid capture of three-dimensional data also have to be investigated. The area can be improved by increased use of artificial intelligence to automate a lot more of the processes of data capture process. This research has seen that the technology and knowledge for distributed and higher dimensional geographic information does exist in the scientific community. What is required is for GIS researchers to examine technologies outside the area of geomatics and understand how these can aid in the analysis and display of spatial information.

The research into techniques for single images shows that enhancements can be made to the Phoenix approach to increase accuracy. The accuracy of the model is influenced by the selection of the initial control points, as shown by the tests on the location of vanishing points in Section 9.2.1. Various methods can be used to increase this selection, for example weighting of the construction lines and statistical analysis of the intersection points. Further to these, artificial intelligence research has developed automated techniques for the creation of the vanishing points of an image, as discussed in Section 7.5. Systems such as Canoma further aid the user by allowing them to create geometric primitives on the image that can then be translated to register with the objects to be created.

Due to the speed at which technology is progressing, it is impossible to predict where computing will have taken geographic information in five years time. Increased reduction of cost will see many technologies, for example VR hardware and multi processors, moving from industry and the military to the desktop. The boundaries of current hardware and software are being pushed and exceeded all the time, with faster and more powerful versions being released and developed. Currently, GIS developers are waiting for the acceptance of advanced technologies before using them. The speed that computer technology progresses requires incorporation of these ideas as they develop.

References

- 3D Scanners, (1998), *3D Scanners – home page*, <http://www.3dscanners.com/>, accessed 07/01/01
- Abbot, E.A., (1884), *Flatland: A Romance of Many Dimensions*, Seeley and Co. Ltd., London 96p, (republished: Dover, 1992)
- AGOCG, (1998), *Collaborative Working and Learning*, <http://www.agocg.ac.uk>, accessed 06/01/00
- Albrecht, J. and Kemppainen, H., (1996), A Framework for Defining the new ISO Standard for Spatial Operators, *Proceedings of GISRUUK '96 Conference*, Canterbury, pp78-84
- ANSI, (1998), SQL3 Object Model, *ANSI X3H7 Object Model Feature Matrix*, <http://www.objs.com/x3h7/sql3.htm>, accessed 01/10/98
- Barnard, S.T., (1983), Interpreting Perspective Images, *Artificial Intelligence*, Vol. 21, pp435-462
- Berry, J.K., Buckley, D. and Ulbricht, C., (1998) Visualize Realistic Landscapes, *GIS World*, August 1998, pp42-47
- Bissell, A. and Papageorigiou, P., (1996), Applications of a Fast Voxel Renderer in Medical and Scientific Visualisation, *Eurographics '96*, available <http://www.voxar.com/techinfo/papers/>, accessed 10/11/98
- Bottrill, A., Pearson, G., Taylor, G., Parsley, S. and Thomas, P., (1998), 3D modelling of structures - new v. traditional methods, *Surveying World*, Vol. 6, No. 5, pp22-25
- Bourdakis, V., (1996), From CAAD to VR; Building a VRML model of London's West End, *Proceedings of The Third UK VR-SIG Conference*, De Montfort University, Leicester, 3rd July 1996
- Canadian Department of National Defence, (1988), *Aide-memoire of Graphical Techniques for the Rectification of Terrestrial Single Frame Photography*, National Defence Headquarters, Ottawa, 64p.
- Capes, R., (1998), Developments in Earth Observation - how new space-borne sensors will affect mapping, *Surveying World*, September/October 1998, pp24-27
- CASA, (1998), *CASA – home page*, <http://www.casa.ucl.ac.uk/>, accessed 23/07/98
- Charitos and Rutherford, (1996), Guidelines for the Design of Virtual Environments, *Proceedings of The Third UK VR-SIG Conference*, De Montfort University, Leicester, 3rd July 1996
- Clowes, M.B., (1971), On seeing things, *Artificial Intelligence*, Vol. 2, No. 1, pp79-116

- Collinson, A., (1997), Virtual Worlds, *The Cartographic Journal*, Vol. 34, No. 2, pp117-124
- Cooper, M.A.R. and Robson, S., (1996), Theory of close range photogrammetry, Chapter 7 in *Close Range Photogrammetry and Machine Vision*, (ed. Atkinson, K.), Whittles Publishing, Latheronwheel, Scotland, pp9-51.
- Davies, C., (1994), GIS: The View From the Front Line, *Mapping Awareness*, April 1994, pp12-13
- Debevec, P.E., Taylor, C.J. and Malik, J., (1996), Modelling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach, *Conference proceedings of SIGGRAPH '96* pp11-20.
- Doneus, M., (1996), Introduction to Photogrammetry, <http://www.univie.ac.at/Luftbildarchiv/wgv/into.htm>, accessed 15/07/98
- Ereno, F., (1998), *Introduction to Object-Oriented GIS Technology*, <http://stss-www.jrc.it/people/federico/OOgis/index.htm>, accessed 07/10/98
- Erricker, B.C. (1976), *Advanced General Statistics*, Hodder and Stoughton Ltd, London.
- Eyre, J., (1998), *Fully Immersive Spherical Projection System (The Cybersphere)*, <http://www.ndirect.co.uk/~vr-systems/sphere1.htm>, accessed 02/11/98
- Fairbairn, D., (1987), *The Creation of Perspective Views: An Example Using Ordnance Survey Data*, MSc Thesis, Department of Surveying, University of Newcastle upon Tyne, April 1987
- Fairbairn, D. and Parsley, S., (1997), The Use of VRML for Cartographic Presentation, *Computers and Geosciences*, Vol. 23, No. 4, pp475-481
- Faust, N.L., (1993), The Virtual Reality of GIS, *Environment and Planning B: Planning and Design*, 22, pp257-268
- Flood, M. and Gutelius, B., (1997), Commercial Implications of Topographic Terrain Mapping Using Scanning Airborne Laser Radar, *Photogrammetric Engineering and Remote Sensing*, Vol. LXIII, No. 4, pp327-329 and pp363-366
- GeoInformation International, (1998), *Cities Revealed – home page*, <http://www.crworld.co.uk/>, accessed 06/11/98
- Ghormley, K., (1995), Puzzling Out the Next-Generation GIS, *GIS World*, October 1995, pp50-52
- Gillings, M. and Goodrick, G.T., (1996), Sensuous and Reflexive GIS: Exploring Visualisation and VRML, *Internet Archaeology*, Issue 1, <http://intarch.ac.uk/journal/issue1/gillings/index.html>, accessed 11/01/96
- Goodchild, M.F., (1987), A Spatial Analytical Perspective as Geographical Information Systems, *IJGIS*, Vol. 1, No. 4, pp327-324

- Hacklay, M., (1998), A Survey of Current Trends in Incorporating Virtual Reality and Geographical Information Systems, *Proceedings of the GISRUUK '98 Conference*, Edinburgh, Session 10-6
- Hausmann, B., (1994), Visualization of Regular Polytopes in Three and Four Dimensions, *Computer Graphics Forum*, Vol. 13, No.3, p C-305
- Havercroft, M., (1997), *Positional Accuracy of large-scale data and products*, <http://www.ordsvy.gov.uk/literatu/infopapr/1997/cons0397.html>, accessed 06/11/98
- Hearn, D. and Baker, M.P., (1993), *Computer Graphics*, Prentice Hall
- Hoffman, D.A., (1971), Impossible objects as nonsense sentences, *Machine Intelligence*, Vol. 6
- Homer, A. and Ullman, C., (1997), *Instant IE4 Dynamic HTML*, Wrox Press, Birmingham, 576p
- Hoss, H, (1996), DTM Derivation with Laser Scanner Data, *Geomatics Info Magazine International*, Vol. 10
- InSpeck, (1998), *InSpeck-3D: Full Color and Black & White Non-Contact Optical Three-Dimensional Digitizers*, <http://www.pasociety.org/hv/local/inspeck/>, accessed 10/11/98
- Kanade, T., (1981), Recovery of the Three-Dimensional Shape of an Object from a Single View, *Artificial Intelligence*, Vol. 17, pp409-460
- Kanji, G. K. (1995), *100 Statistical Tests*, SAGE Publications Ltd, London.
- Keates, J.S., (1989), *Cartographic Design and Production*, Second Edition, Longman Scientific and Technical, 254p
- Kemp, Z., (1995), Multimedia and Spatial Information Systems, *IEEE Multimedia*, Vol. 2, No. 4, pp68-76
- Kite, T., (1998), *Universal Data Access: It's Not Your Father's Data Anymore*, http://www.eu.microsoft.com/msdn/news/feature/datajul98/da_intro.htm, accessed 14/07/98
- Kraak, M.J., (1993), Cartographic Terrain Modeling in a Three Dimensional GIS Environment, *Cartography and Geographic Information Systems*, Vol. 20, No. 1, pp13-18
- Loeb, V., (2000), U.S. is Relaxing Rules on Sale of Satellite Photos, *Washington Post*, 16th December 2000, pA03, <http://washingtonpost.com/wp-dyn/articles/A14261-2000Dec15.html>, accessed 30/12/00

- Lombardo, D. and Kemp, Z., (1997), Towards a Model for Multimedia and its Application to Geographical Information Systems, *Innovations in GIS 4*, (ed. Kemp, Z.), Taylor & Francis, Chichester, pp56-69
- Longuet-Higgins, H.C., (1981), A Computer Algorithm for Reconstructing a Scene From Two Projections, *Nature*, Vol. 293, pp133-135
- MacEachren, A.M., and Taylor, D.R.F., (1994), *Visualisation in Modern Cartography*, Elsevier Science, New York, 345p
- Marshall, D., (1997), *Methods of Acquisition*,
http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/MARSHALL/node12.html, accessed 21/10/98
- McCleary, G.F. and Jenks, G.F., (1993), Cartography and Map Displays, *Pictorial Communication in Virtual and Real Environments*, 2nd Edition, Ed. Ellis, R., Taylor and Francis
- McKee, L., (1998), Open for Business, *Mapping Awareness*, February 1998, pp18-20
- McLaren, R.A. and Kennie, T.J.M., (1989), Visualisation of Digital Terrain Models: Techniques and Applications, *Three Dimensional Applications in Geographic Information Systems*, Ed. Raper, J., Taylor and Francis
- Microsoft, (1998a), *Dr. GUI on components, COM and ATL*,
<http://www.eu.microsoft.com/msdn/news/drgui/020298.htm>, accessed 14/07/98
- Microsoft, (1998b), *Microsoft Universal Data Access*, <http://www.microsoft.com/data/>, accessed 07/10/98
- Microsoft, (1998c), *Windows DNA: Building Windows Applications for the Internet Age*, <http://www.microsoft.com/dna/overview/dnawp.asp>, accessed 18/10/98
- Mitchell, W.L. and Pendlebury, M., (1996), Reconstruction of the Egyptian Tomb of Menna using VRML, *Proceedings of The Third UK VR-SIG Conference*, De Montfort University, Leicester, 3rd July 1996
- Mullins, J., (2000), Integrated circuits, Inside Science136, *New Scientist*, 9th December 2000, No. 2268, p3
- NCGIA, (1990a), History of GIS, *NCGIA Core Curriculum*, Unit 23,
<http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u23.html#UNIT23>, accessed 28/04/98
- NCGIA, (1990b), Hierarchical Data Structures, *NCGIA Core Curriculum*, Unit 36,
<http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u36.html#UNIT36>, accessed 11/01/01
- Neves, J.G., Goncalves, J.M. and Silva, J.P., (1998), Interfacing Spatial Information in Virtual Environments, <http://virtual.dcea.fct.unl.pt/gasa/papers/vgis/>, accessed 22/06/98

- Newsam, G., (1996), *SolidFit - Shape and Measurements from Perspective Distortion*, <http://www.cssip.edu.au/~vision/solidFit.html>, accessed 05/11/96
- OGC, (1998), *The Open GIS Consortium – home page*, <http://www.opengis.org>, last accessed 13/10/98
- ODMG, (1998), *The Object Data Management Group – home page*, <http://www.odmg.org>, last accessed 13/10/98
- OMG, (1998), *The Object Management Group – home page*, <http://www.omg.org>, last accessed 13/10/98
- Parodi, P. and Torre, V., (1994), On the Complexity of Labeling Perspective Projections of Polyhedral Scenes, *Artificial Intelligence*, Vol. 70, pp239-276
- Parsley, S. and Taylor, G. (1998) Methodologies and tools for large scale, real world virtual environments: a prototype to enhance the understanding of historic Newcastle upon Tyne, *Chapter 7 in Innovations in GIS 5* (ed.Carver, S.), Taylor & Francis, Chichester.
- Peucker, T., (1980), The Impact of Different Mathematical Approaches to Contouring, *Cartographica*, Vol. 17, No. 2, pp73-95
- Quest, (1996), *Differential GPS*, <http://www.qualityeng.co.uk/dgps/dgps3.html>, accessed 05/11/98
- Raper, J., (1994), *Progress in Spatial Multimedia*, Position paper for the GISDATA Specialist meeting on GIS and Multimedia, Rostock, May 1994
- Rhind, D., (1995), Ordnance Survey to the Millenium: the new polocies and plans of the National Mapping Agency, *Proceedings of the AGI '95 conference*, available <http://www.ordsvy.gov.uk/literatu/external/agi95/htagir.html>, accessed 01/10/98
- Scheller, S., (1997), *Hypercube Animator*, <http://cip.physik.uni-bonn.de/ScienceSite/hypercubus/animator>, accessed 14/10/98
- Schmalstieg, D., (1997), *LODestar: An Octree-Based Level of Detail Generator for VRML*, <http://www.cg.tuwien.ac.at/~dieter/>, accessed 15/10/98
- Schmidt, D.C., (1998), *Distributed Object Computing with CORBA*, <http://www.cs.wustl.edu/~schmidt/corba.html>, accessed 07/10/98
- Seife, C., (1999), Tiff Over GIFs Sparks Net Uproar, *New Scientist*, 11th September 1999, No 2203, p7
- Seymour, J., (1996), Virtually Real, Really Sick, *New Scientist*, 27th January 1996, pp34-37
- Sinning-Meister, N., Gruen, A. and Dan, H., (1996), 3D City Models for CAAD-Supported Analysis and Design of Urban Areas, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 51, pp196-208

- Smyth, M.M., Collins, A.F., Morris, P.E. and Levy, P., (1994), *Cognition in Action*, Lawrence Earlbaum Associates, Hove
- Space Imaging, (1998), *Space Imaging – home page*, <http://www.spaceimaging.com/>, accessed 11/01/01
- Stewart, I., (1995), *Natures Numbers*, Harper Collins, 164p
- Taylor, D.R.F., (1991), A Conceptual Basis for Cartography: New Directions for the Cartographic Era, *Cartographic Journal*, Vol. 28, pp213-216
- Taylor, G., Fairbairn, D. J. and Parsley, S., (2000), The Phoenix Algorithm for Three Dimensional Data Capture, *Photogrammetric Record*, 16(96), October 2000, pp1007-1021
- Thoen, B., (1995), Interactive Mapping and GIS Thrive on the Web, *GIS World*, No. 5859, October 1995
- Tidwell, M., (1996), *A Virtual Retinal Display for Augmenting Ambient Visual Environments*, Masters Thesis, available <http://www.hitl.washington.edu/publications/tidwell/ch1.html>, accessed 21/05/96
- Tricorder, (1998), *Tricorder – home page*, <http://www.tricorder.co.uk/>, accessed 28/10/98
- Turner, A.K., (1995), Three-dimensional visualisation of geo-science data: status, problems, and potentials, *ITC Journal*, 2, pp151-152
- Turner, A.K., (1998), Powerful PCs and Virtual Worlds Bolster Geoscientific Visualisation, *GIS World*, July 1998, pp36-38
- UCAID, (2001), *Internet2 Website*, <http://www.internet2.edu/>, accessed 12/01/01
- Virage, (1998), *Video Cataloger 1.3*, <http://www.virage.com/market/cataloger-body.html>, accessed 29/10/98
- von Wyss, M., (1996), The Production of Smooth Scale Changes in an Animated Map Project, *Cartographic Perspectives*, No. 23, pp12-20
- Voxar, (1998), *Voxar Technologies – home page*, <http://www.voxar.com>, accessed 10/11/98
- Worboys, M.F. and Hamilton, G.W., (1996), User Perspectives on Geographic Space, *Proceedings of the GISRUUK '96 Conference*, Canterbury, pp151-155
- Zhao, X. and Sun, J., (1994), Reconstruction of a Symmetrical Object from its Perspective Image, *Computers and Graphics*, Vol. 18, No. 4, pp463-467
- ZDNet, (1998), *Computing Encyclopedia*, <http://www.zdnet.com/encyclopedia>, accessed 29/10/98

Table of Appendices

APPENDIX 1: SURVEY DATA	231
FIGURE A10.1: PLAN VIEW OF DETAIL POINTS FROM SECOND SURVEY	240
TABLE A1: RESULTS OF THEODOLITE SURVEY AND PHOENIX APPROACH	241
TABLE A2: DIFFERENCES IN DISTANCE BETWEEN SURVEY POINTS	242
TABLE A3: PERCENTAGE DIFFERENCES IN DISTANCE BETWEEN SURVEY POINTS	243

This section contains the data from the survey of the Police House as discussed in Section 9.3. The data is included here to allow any future analysis of the results to be carried out if required.

Appendix 1: Survey Data

STAR*NET Adjustment Program
Copyright 1991 STARPLUS SOFTWARE, INC.
Licensed for Use by University of Newcastle Upon Tyne
Serial Number 10236

STAR*NET V4.066
Run Date : Sat Jul 03 16:35:52 1993

Summary of Files Used

Input data file :D:\RESEARCH\SURVEY\COMPLETE.DAT
Output listing (this file) : D:\RESEARCH\SURVEY\COMPLETE.LST
Coordinates : D:\RESEARCH\SURVEY\COMPLETE.PTS
Project parameters : D:\RESEARCH\SURVEY\COMPLETE.PRJ
Error log : D:\RESEARCH\SURVEY\COMPLETE.ERR
Plot File : D:\RESEARCH\SURVEY\COMPLETE.SPL

Adjustment Options

STAR*NET Run Mode : Adjust Only
Type of Adjustment : 3D
Input Order for Coordinates : X,Y,Z
Coordinate System : LOCAL
Project Scale Factor : 1.00000000
Input Lineal Units : Meters
Input Order for Angle Stations : At-From-To
Maximum Number of Iterations : 10
Convergence Limit Test Value : 0.010000

Correct Zeniths for Curvature & Refraction: No
Adjust 3D Observations for Vert Divergence: No
3D Data Input Mode : Slope/Zenith

Listing Options

Print Copy of Input Data File : No
Print Summary of All Input Observations : Yes
Print Coordinate Changes Each Iteration : No
Print Horiz Traverse Closure Summaries : Yes

Default Instrument Settings

Default Standard Error for Distances : 0.03000
Default Standard Error for Angles : 4.000
Default Standard Error for Directions : 4.000
Default Standard Error for Az/Bearings : 4.000
Default Standard Error for Zeniths : 20.00
Default Standard Error for Delta Elev : 0.05000

Centering Error for Instrument : 0.0000
Centering Error for Target : 0.0000
EDM Parts Per Million : 0.0000

Summary of All Unadjusted Input Observations

=====

Number of Stations with Coordinates = 38

Fixed Stations	X	Y	Z	Description
S1	424816.850	565082.540	49.000	
Free Stations	X	Y	Z	Description
S2	424812.559	565066.237	48.891	
S3	424824.650	565052.399	47.634	
P1	424843.000	565083.000	52.000	

P2	424838.000	565078.000	52.000
P3	424838.000	565079.000	57.000
P4	424837.000	565078.000	57.000
P5	424837.000	565078.000	59.000
P6	424836.000	565077.000	59.000
P7	424836.000	565077.000	58.000
P8	424828.000	565071.000	58.000
P9	424828.000	565071.000	49.000
P10	424833.000	565075.000	52.000
P11	424832.000	565074.000	52.000
P12	424832.000	565074.000	51.000
P13	424833.000	565075.000	51.000
P17	424838.000	565060.000	58.000
P18	424838.000	565060.000	49.000
P21	424829.000	565071.000	49.000
P22	424829.000	565071.000	52.000
P23	424830.000	565069.000	52.000
P24	424830.000	565069.000	49.000
P25	424829.000	565070.000	49.000
P26	424829.000	565070.000	52.000
P27	424829.000	565070.000	52.000
P28	424829.000	565070.000	49.000
P29	424830.000	565069.000	50.000
P30	424830.000	565069.000	52.000
P31	424831.000	565068.000	52.000
P32	424831.000	565068.000	50.000
P33	424830.000	565069.000	52.000
P34	424831.000	565068.000	52.000
P35	424832.000	565067.000	50.000
P36	424832.000	565067.000	52.000
P37	424832.000	565066.000	52.000
P38	424832.000	565066.000	50.000
P39	424832.000	565067.000	52.000
P40	424832.000	565066.000	52.000

Number of Angle Observations = 73

At	From	To	Obs Angle	StdErr
S1	S3	S2	29-15-13.00	4.00
S2	S1	S3	124-08-43.00	4.00
S3	S2	S1	26-36-10.00	4.00
S1	S2	P1	253-18-19.00	4.00
S1	S2	P2	264-38-31.00	4.00
S1	S2	P3	264-30-57.00	4.00
S1	S2	P4	267-28-34.00	4.00
S1	S2	P5	267-34-33.00	4.00
S1	S2	P6	270-21-52.00	4.00
S1	S2	P7	270-07-23.00	4.00
S1	S2	P10	279-12-40.00	4.00
S1	S2	P11	282-43-51.00	4.00
S1	S2	P12	282-44-18.00	4.00
S1	S2	P13	279-13-03.00	4.00
S2	S1	P1	46-15-23.00	4.00
S2	S1	P2	48-55-42.00	4.00
S2	S1	P3	48-54-33.00	4.00
S2	S1	P4	49-32-43.00	4.00
S2	S1	P5	49-36-33.00	4.00
S2	S1	P6	50-12-54.00	4.00
S2	S1	P7	50-00-54.00	4.00
S2	S1	P8	58-08-58.00	4.00
S2	S1	P9	58-05-59.00	4.00
S2	S1	P10	52-15-09.00	4.00
S2	S1	P11	53-10-31.00	4.00
S2	S1	P12	53-11-01.00	4.00
S2	S1	P13	52-15-43.00	4.00
S2	S1	P17	88-53-33.00	4.00
S2	S1	P18	88-48-28.00	4.00
S2	S1	P21	58-48-14.00	4.00
S2	S1	P22	58-51-52.00	4.00
S2	S1	P23	63-23-21.00	4.00
S2	S1	P24	63-22-22.00	4.00
S2	S1	P25	59-39-50.00	4.00
S2	S1	P26	59-37-25.00	4.00
S2	S1	P27	62-36-29.00	4.00

S2	S1	P28	62-33-32.00	4.00
S2	S1	P29	66-01-14.00	4.00
S2	S1	P30	66-01-26.00	4.00
S2	S1	P31	68-48-08.00	4.00
S2	S1	P32	68-50-06.00	4.00
S2	S1	P33	65-38-14.00	4.00
S2	S1	P34	69-14-59.00	4.00
S2	S1	P35	71-47-08.00	4.00
S2	S1	P36	71-46-58.00	4.00
S2	S1	P37	74-14-23.00	4.00
S2	S1	P38	74-16-00.00	4.00
S2	S1	P39	71-23-38.00	4.00
S2	S1	P40	74-39-04.00	4.00
S3	S2	P8	53-54-29.00	4.00
S3	S2	P9	54-01-18.00	4.00
S3	S2	P17	102-00-33.00	4.00
S3	S2	P18	101-56-35.00	4.00
S3	S2	P21	54-21-34.00	4.00
S3	S2	P22	54-22-30.00	4.00
S3	S2	P23	58-12-45.00	4.00
S3	S2	P24	58-15-51.00	4.00
S3	S2	P25	55-04-20.00	4.00
S3	S2	P26	55-02-58.00	4.00
S3	S2	P27	57-33-28.00	4.00
S3	S2	P28	57-31-53.00	4.00
S3	S2	P29	61-09-23.00	4.00
S3	S2	P30	60-57-13.00	4.00
S3	S2	P31	63-59-50.00	4.00
S3	S2	P32	64-02-42.00	4.00
S3	S2	P33	60-27-50.00	4.00
S3	S2	P34	64-28-02.00	4.00
S3	S2	P35	67-41-29.00	4.00
S3	S2	P36	67-41-29.00	4.00
S3	S2	P37	71-06-00.00	4.00
S3	S2	P38	71-08-33.00	4.00
S3	S2	P39	67-09-24.00	4.00
S3	S2	P40	71-40-54.00	4.00

Number of Distance Observations = 6
Distance Units Are Meters

From	To	Obs Dist	StdErr	HI-HT	Flags
S1	S2	16.8580	0.030	-0.103	S
S2	S3	18.3990	0.030	-0.025	S
S3	S1	31.1590	0.030	0.128	S
S1	S3	31.1600	0.030	-0.128	S
S2	S1	16.8580	0.030	0.103	S
S3	S2	18.3990	0.030	0.025	S

Number of Zenith Observations = 69

From	To	Obs Zenith	StdErr	HI-HT	Flags
S1	S2	90-01-17.00	20.00	-0.103	
S2	S3	93-50-17.00	20.00	-0.025	
S3	S1	87-43-05.00	20.00	0.128	
S1	S3	92-16-53.00	20.00	-0.128	
S2	S1	89-58-30.00	20.00	0.103	
S3	S2	86-09-34.00	20.00	0.025	
S1	P1	86-29-41.00	20.00	1.371	
S1	P2	85-34-31.00	20.00	1.371	
S1	P3	71-59-08.00	20.00	1.371	
S1	P4	71-08-56.00	20.00	1.371	
S1	P5	66-01-38.00	20.00	1.371	
S1	P6	65-19-04.00	20.00	1.371	
S1	P7	67-37-20.00	20.00	1.371	
S1	P8	63-23-15.00	20.00	1.371	
S1	P9	93-05-25.00	20.00	1.371	
S1	P10	84-10-53.00	20.00	1.371	
S1	P11	84-00-13.00	20.00	1.371	
S1	P12	85-28-00.00	20.00	1.371	
S1	P13	85-36-19.00	20.00	1.371	
S1	P17	75-07-50.00	20.00	1.371	
S2	P2	86-37-51.00	20.00	1.474	

S2	P3	76-09-34.00	20.00	1.474
S2	P5	71-16-07.00	20.00	1.474
S2	P6	70-31-57.00	20.00	1.474
S2	P7	72-28-14.00	20.00	1.474
S2	P8	64-11-55.00	20.00	1.474
S2	P10	85-18-47.00	20.00	1.474
S2	P11	85-03-21.00	20.00	1.474
S2	P12	86-15-27.00	20.00	1.474
S2	P13	86-27-32.00	20.00	1.474
S2	P17	72-40-02.00	20.00	1.474
S2	P21	92-58-49.00	20.00	1.474
S2	P22	82-56-32.00	20.00	1.474
S2	P23	83-13-50.00	20.00	1.474
S2	P24	92-52-08.00	20.00	1.474
S2	P25	92-54-36.00	20.00	1.474
S2	P26	83-50-19.00	20.00	1.474
S2	P27	84-00-11.00	20.00	1.474
S2	P28	92-53-09.00	20.00	1.474
S2	P30	84-06-48.00	20.00	1.474
S2	P31	84-18-15.00	20.00	1.474
S2	P32	90-23-23.00	20.00	1.474
S2	P33	83-08-44.00	20.00	1.474
S2	P34	83-24-55.00	20.00	1.474
S2	P35	90-21-00.00	20.00	1.474
S2	P36	84-30-07.00	20.00	1.474
S2	P37	84-43-09.00	20.00	1.474
S2	P39	83-34-56.00	20.00	1.474
S2	P40	83-54-45.00	20.00	1.474
S3	P8	63-50-15.00	20.00	1.499
S3	P9	88-58-33.00	20.00	1.499
S3	P18	89-31-17.00	20.00	1.499
S3	P23	79-37-51.00	20.00	1.499
S3	P24	88-56-15.00	20.00	1.499
S3	P26	80-45-23.00	20.00	1.499
S3	P27	80-29-27.00	20.00	1.499
S3	P28	88-56-54.00	20.00	1.499
S3	P29	86-24-53.00	20.00	1.499
S3	P30	80-04-04.00	20.00	1.499
S3	P31	79-47-06.00	20.00	1.499
S3	P32	86-21-21.00	20.00	1.499
S3	P33	79-09-27.00	20.00	1.499
S3	P34	78-45-21.00	20.00	1.499
S3	P35	86-13-14.00	20.00	1.499
S3	P36	79-28-21.00	20.00	1.499
S3	P37	79-16-11.00	20.00	1.499
S3	P38	86-10-02.00	20.00	1.499
S3	P39	78-30-30.00	20.00	1.499
S3	P40	78-13-00.00	20.00	1.499

Number of Azimuth/Bearing Observations = 1

From	To	Azimuth	StdErr
S2	S1	14-44-43.00	FIXED

Network Solution
=====

Solution Has Converged in 5 Iterations

Final Results
=====

Updated Coordinates and Changes from Last Iteration

Station	X	Y	Z	dX	dY	dZ
S1	424816.850	565082.540	49.000	-0.000	-0.000	0.000
S2	424812.563	565066.252	48.890	-0.000	0.000	-0.000
S3	424824.648	565052.403	47.632	-0.000	-0.000	-0.000

P1	424843.592	565083.450	52.010	-0.000	0.000	-0.000
P2	424838.311	565078.992	52.055	-0.000	-0.000	0.000
P3	424838.369	565079.031	57.460	-0.000	-0.000	-0.000
P4	424837.227	565078.126	57.489	-0.000	0.000	-0.000
P5	424837.222	565078.090	59.642	0.000	-0.000	-0.000
P6	424836.229	565077.307	59.597	0.000	0.000	0.000
P7	424836.211	565077.400	58.620	-0.000	0.000	0.000
P8	424828.939	565071.291	58.646	-0.000	0.000	0.000
P9	424828.985	565071.321	49.478	0.000	0.000	0.000
P10	424833.503	565075.141	52.228	0.000	0.000	0.000
P11	424832.568	565074.366	52.232	0.000	-0.000	0.000
P12	424832.568	565074.363	51.776	0.000	-0.000	-0.000
P13	424833.505	565075.138	51.772	-0.000	-0.000	0.000
P17	424838.307	565060.006	58.632	0.000	0.000	0.000
P18	424838.331	565060.040	49.261	-0.000	-0.000	-0.000
P21	424829.056	565071.122	49.469	0.000	0.000	-0.000
P22	424829.057	565071.103	52.493	-0.000	0.000	0.000
P23	424830.039	565069.923	52.484	0.000	-0.000	0.000
P24	424830.059	565069.933	49.469	-0.000	-0.000	0.000
P25	424829.249	565070.908	49.483	0.000	0.000	0.000
P26	424829.244	565070.919	52.235	-0.000	0.000	0.000
P27	424829.884	565070.138	52.228	-0.000	0.000	0.000
P28	424829.879	565070.153	49.470	-0.000	-0.000	-0.000
P29	424830.782	565069.214	50.252	0.000	-0.000	-0.000
P30	424830.710	565069.201	52.259	0.000	0.000	0.000
P31	424831.394	565068.381	52.255	0.000	0.000	-0.000
P32	424831.405	565068.372	50.235	0.000	0.000	0.000
P33	424830.585	565069.306	52.562	-0.000	0.000	0.000
P34	424831.489	565068.243	52.561	-0.000	0.000	0.000
P35	424832.172	565067.440	50.242	0.000	-0.000	-0.000
P36	424832.173	565067.441	52.255	-0.000	-0.000	0.000
P37	424832.848	565066.611	52.239	-0.000	-0.000	-0.000
P38	424832.857	565066.602	50.229	-0.000	0.000	-0.000
P39	424832.060	565067.567	52.562	0.000	-0.000	0.000
P40	424832.956	565066.467	52.538	0.000	-0.000	0.000

Statistical Summary
=====

Number of Observations = 149
Number of Unknowns = 111
Degrees of Freedom = 38

Data Type	Count	Weighted Residuals	Error Factor
Distances	6	1.70	1.05
Angles	73	0.77	0.20
Directions	0	0.00	0.00
Az/Bearings	1	0.00	0.00
Stations	38	0.00	0.00
Zeniths	69	8.69	0.70
Delta Elev	0	0.00	0.00
Total	187	11.16	0.54

Adjustment passes the Chi Square test at 5% level

Adjusted Observations and Residuals
=====

Adjusted Angle Observations

At	From	To	Adj Angle	Residual	StdErr	StdRes
S1	S3	S2	29-15-10.72	-0-00-02.28	4.00	0.6
S2	S1	S3	124-08-40.98	-0-00-02.02	4.00	0.5
S3	S2	S1	26-36-08.30	-0-00-01.70	4.00	0.4
S1	S2	P1	253-18-19.00	0-00-00.00	4.00	0.0
S1	S2	P2	264-38-31.00	-0-00-00.00	4.00	0.0
S1	S2	P3	264-30-57.01	0-00-00.01	4.00	0.0

S1	S2	P4	267-28-34.00	0-00-00.00	4.00	0.0
S1	S2	P5	267-34-33.01	0-00-00.01	4.00	0.0
S1	S2	P6	270-21-52.00	0-00-00.00	4.00	0.0
S1	S2	P7	270-07-23.00	0-00-00.00	4.00	0.0
S1	S2	P10	279-12-40.00	0-00-00.00	4.00	0.0
S1	S2	P11	282-43-51.00	-0-00-00.00	4.00	0.0
S1	S2	P12	282-44-18.00	0-00-00.00	4.00	0.0
S1	S2	P13	279-13-03.00	0-00-00.00	4.00	0.0
S2	S1	P1	46-15-23.00	0-00-00.00	4.00	0.0
S2	S1	P2	48-55-42.01	0-00-00.01	4.00	0.0
S2	S1	P3	48-54-32.92	-0-00-00.08	4.00	0.0
S2	S1	P4	49-32-43.00	0-00-00.00	4.00	0.0
S2	S1	P5	49-36-32.84	-0-00-00.16	4.00	0.0
S2	S1	P6	50-12-54.05	0-00-00.05	4.00	0.0
S2	S1	P7	50-00-54.14	0-00-00.14	4.00	0.0
S2	S1	P8	58-08-58.22	0-00-00.22	4.00	0.1
S2	S1	P9	58-05-59.00	0-00-00.00	4.00	0.0
S2	S1	P10	52-15-09.01	0-00-00.01	4.00	0.0
S2	S1	P11	53-10-30.99	-0-00-00.01	4.00	0.0
S2	S1	P12	53-11-01.00	0-00-00.00	4.00	0.0
S2	S1	P13	52-15-43.00	0-00-00.00	4.00	0.0
S2	S1	P17	88-53-32.97	-0-00-00.03	4.00	0.0
S2	S1	P18	88-48-28.00	0-00-00.00	4.00	0.0
S2	S1	P21	58-48-14.00	0-00-00.00	4.00	0.0
S2	S1	P22	58-51-52.00	0-00-00.00	4.00	0.0
S2	S1	P23	63-23-21.03	0-00-00.03	4.00	0.0
S2	S1	P24	63-22-22.03	0-00-00.03	4.00	0.0
S2	S1	P25	59-39-50.00	0-00-00.00	4.00	0.0
S2	S1	P26	59-37-25.04	0-00-00.04	4.00	0.0
S2	S1	P27	62-36-28.98	-0-00-00.02	4.00	0.0
S2	S1	P28	62-33-32.01	0-00-00.01	4.00	0.0
S2	S1	P29	66-01-14.00	-0-00-00.00	4.00	0.0
S2	S1	P30	66-01-25.94	-0-00-00.06	4.00	0.0
S2	S1	P31	68-48-08.06	0-00-00.06	4.00	0.0
S2	S1	P32	68-50-06.00	-0-00-00.00	4.00	0.0
S2	S1	P33	65-38-14.02	0-00-00.02	4.00	0.0
S2	S1	P34	69-14-59.00	-0-00-00.00	4.00	0.0
S2	S1	P35	71-47-07.95	-0-00-00.05	4.00	0.0
S2	S1	P36	71-46-58.02	0-00-00.02	4.00	0.0
S2	S1	P37	74-14-23.00	-0-00-00.00	4.00	0.0
S2	S1	P38	74-16-00.00	0-00-00.00	4.00	0.0
S2	S1	P39	71-23-38.00	-0-00-00.00	4.00	0.0
S2	S1	P40	74-39-03.95	-0-00-00.05	4.00	0.0
S3	S2	P8	53-54-28.86	-0-00-00.14	4.00	0.0
S3	S2	P9	54-01-18.01	0-00-00.01	4.00	0.0
S3	S2	P17	102-00-33.01	0-00-00.01	4.00	0.0
S3	S2	P18	101-56-35.00	0-00-00.00	4.00	0.0
S3	S2	P21	54-21-34.00	0-00-00.00	4.00	0.0
S3	S2	P22	54-22-30.00	0-00-00.00	4.00	0.0
S3	S2	P23	58-12-45.01	0-00-00.01	4.00	0.0
S3	S2	P24	58-15-50.96	-0-00-00.04	4.00	0.0
S3	S2	P25	55-04-20.00	0-00-00.00	4.00	0.0
S3	S2	P26	55-02-58.01	0-00-00.01	4.00	0.0
S3	S2	P27	57-33-28.00	-0-00-00.00	4.00	0.0
S3	S2	P28	57-31-52.98	-0-00-00.02	4.00	0.0
S3	S2	P29	61-09-23.00	0-00-00.00	4.00	0.0
S3	S2	P30	60-57-12.99	-0-00-00.01	4.00	0.0
S3	S2	P31	63-59-50.00	0-00-00.00	4.00	0.0
S3	S2	P32	64-02-42.00	0-00-00.00	4.00	0.0
S3	S2	P33	60-27-50.00	0-00-00.00	4.00	0.0
S3	S2	P34	64-28-02.00	-0-00-00.00	4.00	0.0
S3	S2	P35	67-41-29.03	0-00-00.03	4.00	0.0
S3	S2	P36	67-41-29.00	0-00-00.00	4.00	0.0
S3	S2	P37	71-06-00.00	0-00-00.00	4.00	0.0
S3	S2	P38	71-08-33.00	0-00-00.00	4.00	0.0
S3	S2	P39	67-09-24.00	-0-00-00.00	4.00	0.0
S3	S2	P40	71-40-54.00	0-00-00.00	4.00	0.0

Adjusted Distance Observations [Meters]

At	To	Adj Dist	Residual	StdErr	StdRes
S1	S2	16.8430	-0.0150	0.030	0.5
S2	S3	18.4216	0.0226	0.030	0.8
S3	S1	31.1542	-0.0048	0.030	0.2

S1	S3	31.1542	-0.0058	0.030	0.2
S2	S1	16.8430	-0.0150	0.030	0.5
S3	S2	18.4216	0.0226	0.030	0.8

Adjusted Azimuth/Bearing Observations

From	To	Adj Azimuth	Residual	StdErr	StdRes
S2	S1	14-44-43.00	-0-00-00.00	FIXED	0.0

Adjusted Zenith Observations

From	To	Adj Zenith	Residual	StdErr	StdRes
S1	S2	90-01-25.73	0-00-08.73	20.00	0.4
S2	S3	93-50-21.54	0-00-04.54	20.00	0.2
S3	S1	87-43-04.85	-0-00-00.15	20.00	0.0
S1	S3	92-16-55.15	0-00-02.15	20.00	0.1
S2	S1	89-58-34.27	0-00-04.27	20.00	0.2
S3	S2	86-09-38.46	0-00-04.46	20.00	0.2
S1	P1	86-29-41.00	0-00-00.00	20.00	0.0
S1	P2	85-34-27.49	-0-00-03.51	20.00	0.2
S1	P3	71-59-16.29	0-00-08.29	20.00	0.4
S1	P4	71-08-56.00	0-00-00.00	20.00	0.0
S1	P5	66-01-50.55	0-00-12.55	20.00	0.6
S1	P6	65-18-59.86	-0-00-04.14	20.00	0.2
S1	P7	67-37-07.81	-0-00-12.19	20.00	0.6
S1	P8	63-23-02.55	-0-00-12.45	20.00	0.6
S1	P9	93-05-31.76	0-00-06.76	20.00	0.3
S1	P10	84-10-48.23	-0-00-04.77	20.00	0.2
S1	P11	84-00-14.88	0-00-01.88	20.00	0.1
S1	P12	85-27-58.80	-0-00-01.20	20.00	0.1
S1	P13	85-36-18.67	-0-00-00.33	20.00	0.0
S1	P17	75-07-53.37	0-00-03.37	20.00	0.2
S2	P2	86-37-55.62	0-00-04.62	20.00	0.2
S2	P3	76-09-23.50	-0-00-10.50	20.00	0.5
S2	P5	71-15-51.68	-0-00-15.32	20.00	0.8
S2	P6	70-32-02.00	0-00-05.00	20.00	0.3
S2	P7	72-28-28.96	0-00-14.96	20.00	0.7
S2	P8	64-12-08.99	0-00-13.99	20.00	0.7
S2	P10	85-18-52.93	0-00-05.93	20.00	0.3
S2	P11	85-03-18.72	-0-00-02.28	20.00	0.1
S2	P12	86-15-28.46	0-00-01.46	20.00	0.1
S2	P13	86-27-32.41	0-00-00.41	20.00	0.0
S2	P17	72-39-59.06	-0-00-02.94	20.00	0.1
S2	P21	92-58-49.00	0-00-00.00	20.00	0.0
S2	P22	82-56-32.00	0-00-00.00	20.00	0.0
S2	P23	83-13-44.79	-0-00-05.21	20.00	0.3
S2	P24	92-51-52.52	-0-00-15.48	20.00	0.8
S2	P25	92-54-36.00	0-00-00.00	20.00	0.0
S2	P26	83-50-11.66	-0-00-07.34	20.00	0.4
S2	P27	84-00-15.28	0-00-04.28	20.00	0.2
S2	P28	92-53-03.45	-0-00-05.55	20.00	0.3
S2	P30	84-06-59.16	0-00-11.16	20.00	0.6
S2	P31	84-18-05.22	-0-00-09.78	20.00	0.5
S2	P32	90-23-24.14	0-00-01.14	20.00	0.1
S2	P33	83-08-41.44	-0-00-02.56	20.00	0.1
S2	P34	83-24-55.61	0-00-00.61	20.00	0.0
S2	P35	90-21-16.35	0-00-16.35	20.00	0.8
S2	P36	84-30-04.15	-0-00-02.85	20.00	0.1
S2	P37	84-43-09.58	0-00-00.58	20.00	0.0
S2	P39	83-34-56.36	0-00-00.36	20.00	0.0
S2	P40	83-54-51.48	0-00-06.48	20.00	0.3
S3	P8	63-50-13.57	-0-00-01.43	20.00	0.1
S3	P9	88-58-25.09	-0-00-07.91	20.00	0.4
S3	P18	89-31-17.00	-0-00-00.00	20.00	0.0
S3	P23	79-37-56.45	0-00-05.45	20.00	0.3
S3	P24	88-56-30.85	0-00-15.85	20.00	0.8
S3	P26	80-45-31.20	0-00-08.20	20.00	0.4
S3	P27	80-29-22.47	-0-00-04.53	20.00	0.2
S3	P28	88-56-59.78	0-00-05.78	20.00	0.3
S3	P29	86-24-53.00	-0-00-00.00	20.00	0.0
S3	P30	80-03-52.95	-0-00-11.05	20.00	0.6
S3	P31	79-47-15.15	0-00-09.15	20.00	0.5

S3	P32	86-21-19.95	-0-00-01.05	20.00	0.1
S3	P33	79-09-29.57	0-00-02.57	20.00	0.1
S3	P34	78-45-20.44	-0-00-00.56	20.00	0.0
S3	P35	86-12-59.94	-0-00-14.06	20.00	0.7
S3	P36	79-28-23.50	0-00-02.50	20.00	0.1
S3	P37	79-16-10.52	-0-00-00.48	20.00	0.0
S3	P38	86-10-02.00	-0-00-00.00	20.00	0.0
S3	P39	78-30-29.68	-0-00-00.32	20.00	0.0
S3	P40	78-12-54.65	-0-00-05.35	20.00	0.3

Adjusted Azimuths and Horizontal Distances
=====

From	To	Azimuth	Distance
S1	S2	194-44-43.00	16.8430
S2	S3	138-53-23.98	18.3803
S3	S1	345-29-32.28	31.1295
S1	P1	88-03-02.00	26.7577
S1	P2	99-23-14.00	21.7526
S1	P3	99-15-40.01	21.8029
S1	P4	102-13-17.00	20.8491
S1	P5	102-19-16.01	20.8522
S1	P6	105-06-35.00	20.0734
S1	P7	104-52-06.00	20.0321
S1	P10	113-57-23.00	18.2224
S1	P11	117-28-34.00	17.7158
S1	P12	117-29-01.00	17.7180
S1	P13	113-57-46.00	18.2256
S2	P1	61-00-06.00	35.4767
S2	P2	63-40-25.01	28.7278
S2	P3	63-39-15.92	28.7966
S2	P4	64-17-26.00	27.3733
S2	P5	64-21-15.84	27.3534
S2	P6	64-57-37.05	26.1214
S2	P7	64-45-37.14	26.1443
S2	P8	72-53-41.22	17.1336
S2	P9	72-50-42.00	17.1866
S2	P10	66-59-52.01	22.7483
S2	P11	67-55-13.99	21.5876
S2	P12	67-55-44.00	21.5872
S2	P13	67-00-26.00	22.7490
S2	P17	103-38-15.97	26.4907
S2	P18	103-33-11.00	26.5065
S2	P21	73-32-57.00	17.1966
S2	P22	73-36-35.00	17.1922
S2	P23	78-08-04.03	17.8576
S2	P24	78-07-05.03	17.8793
S2	P25	74-24-33.00	17.3233
S2	P26	74-22-08.04	17.3215
S2	P27	77-21-11.98	17.7517
S2	P28	77-18-15.01	17.7504
S2	P29	80-45-57.00	18.4584
S2	P30	80-46-08.94	18.3853
S2	P31	83-32-51.06	18.9505
S2	P32	83-34-49.00	18.9610
S2	P33	80-22-57.02	18.2791
S2	P34	83-59-42.00	19.0299
S2	P35	86-31-50.95	19.6452
S2	P36	86-31-41.02	19.6458
S2	P37	88-59-06.00	20.2880
S2	P38	89-00-43.00	20.2965
S2	P39	86-08-21.00	19.5410
S2	P40	89-23-46.95	20.3940
S3	P8	12-47-52.84	19.3693
S3	P9	12-54-41.99	19.4087
S3	P17	60-53-56.99	15.6319
S3	P18	60-49-58.98	15.6701
S3	P21	13-14-57.98	19.2304
S3	P22	13-15-53.98	19.2125
S3	P23	17-06-08.99	18.3310
S3	P24	17-09-14.94	18.3460
S3	P25	13-57-43.98	19.0679
S3	P26	13-56-21.99	19.0776

S3	P27	16-26-51.97	18.4918
S3	P28	16-25-16.96	18.5045
S3	P29	20-02-46.98	17.8948
S3	P30	19-50-36.97	17.8582
S3	P31	22-53-13.98	17.3437
S3	P32	22-56-05.98	17.3394
S3	P33	19-21-13.98	17.9149
S3	P34	23-21-25.98	17.2533
S3	P35	26-34-53.00	16.8147
S3	P36	26-34-52.98	16.8158
S3	P37	29-59-23.98	16.4043
S3	P38	30-01-56.98	16.4005
S3	P39	26-02-47.98	16.8786
S3	P40	30-34-17.98	16.3340
S1	P8	132-56-19.01	16.5129
S1	P9	132-45-12.83	16.5265
S1	P17	136-24-10.94	31.1158

Horizontal Traverse Closures of Unadjusted Observations
=====

(Beginning and Ending on ADJUSTED Stations)

TRAVERSE # 1
Begin at S1, Backsight Azimuth to S3 = 165-29-32.28

From	To	Unadj Azimuth	Unadj Dist	Unadj Elev Diff
S1	S2	194-44-45.28	16.8580	-0.1093
S2	S3	138-53-28.28	18.3577	-1.2566
S3	S1	345-29-38.28	31.1343	1.3687

Horizontal Precision	1:3202	66.3500 T	
Misclosures	-0.0194 X	0.0071 Y	0.0028 Elev
Angular Misclosure	0-00-06.00		

Elapsed time = 00:00:24

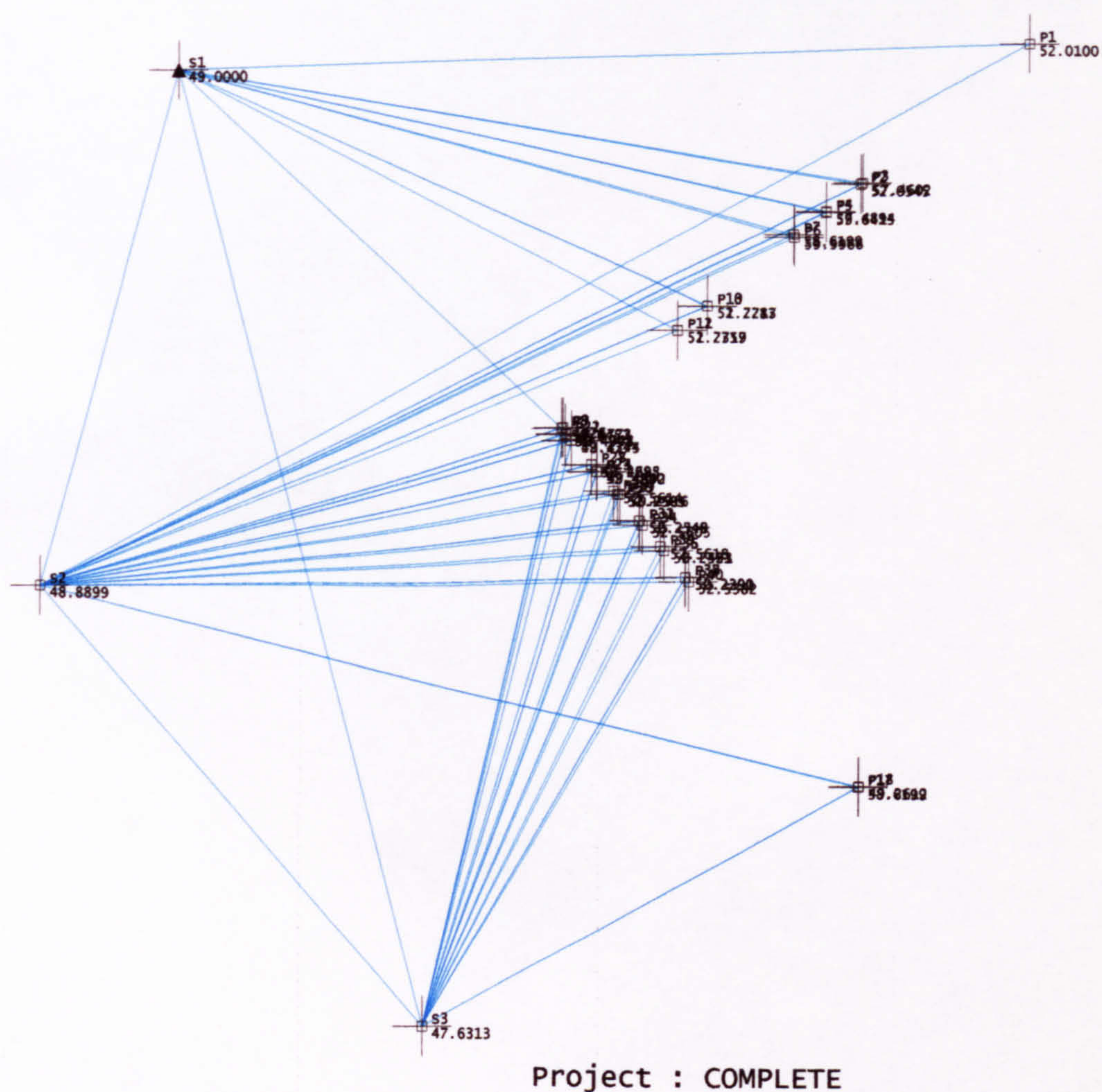


Figure A10.1: Plan view of detail points from second survey

Figure A10.1. The Figure illustrates the plan view of the detail points taken to the building with the lines of sight illustrated in cyan. The three points to the left of the Figure are the control stations.

Table A1: Results of Theodolite survey and Phoenix Approach

	Theodolite Intersection			Phoenix Algorithms			Absolute Difference			dPos (m)
	X	Y	Z	X	Y	Z	dX (m)	dY (m)	dZ (m)	
P1	424843.59	565083.45	52.01	424843.52	565083.41	52.08	0.08	0.04	0.07	0.11
P2	424838.31	565078.99	52.05	424838.00	565078.84	52.19	0.31	0.15	0.13	0.37
P3	424838.37	565079.03	57.46	424837.83	565078.71	57.30	0.54	0.32	0.16	0.65
P4	424837.23	565078.13	57.49	424836.83	565077.88	57.35	0.40	0.24	0.14	0.49
P5	424837.22	565078.09	59.64	424836.70	565077.78	59.44	0.53	0.32	0.20	0.64
P6	424836.23	565077.31	59.60	424835.83	565077.06	59.41	0.40	0.25	0.19	0.51
P7	424836.21	565077.40	58.62	424836.03	565077.22	58.52	0.18	0.18	0.10	0.27
P8	424828.94	565071.29	58.65	424828.83	565071.28	58.45	0.11	0.01	0.19	0.22
P9	424828.99	565071.32	49.48	424829.01	565071.43	49.45	0.02	0.10	0.03	0.11
P10	424833.50	565075.14	52.23	424833.50	565075.14	52.23	0.00	0.00	0.00	0.01
P11	424832.57	565074.37	52.23	424832.56	565074.36	52.23	0.01	0.00	0.00	0.01
P12	424832.57	565074.36	51.78	424832.59	565074.38	51.77	0.02	0.02	0.00	0.03
P13	424833.50	565075.14	51.77	424833.50	565075.14	51.77	0.00	0.00	0.00	0.00
P17	424838.31	565060.01	58.62	424838.52	565059.90	58.53	0.21	0.11	0.09	0.26
P21	424829.06	565071.12	49.47	424829.14	565071.26	49.44	0.09	0.14	0.04	0.17
P22	424829.06	565071.10	52.50	424829.19	565071.20	52.48	0.13	0.10	0.02	0.17
P23	424830.04	565069.92	52.48	424830.19	565069.99	52.46	0.15	0.07	0.03	0.17
P24	424830.06	565069.93	49.47	424830.15	565070.04	49.41	0.09	0.11	0.06	0.15
P25	424829.25	565070.91	49.48	424829.33	565071.03	49.44	0.08	0.12	0.04	0.15
P26	424829.24	565070.92	52.23	424829.37	565070.98	52.20	0.13	0.06	0.03	0.15
P27	424829.88	565070.38	52.23	424830.02	565070.20	52.19	0.13	0.18	0.03	0.23
P28	424829.88	565070.15	49.47	424829.97	565070.25	49.41	0.09	0.10	0.06	0.15
P29	424830.78	565069.21	50.25	424830.76	565069.30	50.23	0.02	0.09	0.03	0.10
P30	424830.71	565069.20	52.26	424830.77	565069.28	52.21	0.06	0.08	0.05	0.12
P31	424831.39	565068.38	52.25	424831.46	565068.45	52.19	0.07	0.07	0.07	0.12
P32	424831.41	565068.37	50.23	424831.47	565068.44	50.19	0.06	0.07	0.05	0.11
P33	424830.59	565069.31	52.56	424830.68	565069.40	52.52	0.10	0.09	0.05	0.14
P34	424831.49	565068.24	52.56	424831.58	565068.31	52.50	0.09	0.06	0.06	0.13
P35	424832.17	565067.44	50.24	424832.22	565067.53	50.18	0.05	0.09	0.06	0.12
P36	424832.17	565067.44	52.26	424832.22	565067.53	52.19	0.05	0.09	0.06	0.12
P37	424832.85	565066.61	52.24	424832.93	565066.68	52.16	0.08	0.07	0.08	0.13
P38	424832.86	565066.60	50.23	424832.94	565066.66	50.15	0.08	0.06	0.08	0.13
P39	424832.06	565067.57	52.56	424832.13	565067.64	52.49	0.07	0.07	0.07	0.13
P40	424832.96	565066.47	52.54	424833.05	565066.53	52.47	0.09	0.06	0.06	0.13

	Theodolite	Phoenix	dDist (m)	Error (%)
P1 - P2	6.911	7.158	0.247	3.6
P2 - P3	5.406	5.117	0.289	5.3
P3 - P4	1.457	1.300	0.158	10.8
P4 - P5	2.152	2.098	0.054	2.5
P5 - P6	1.265	1.125	0.139	11.0
P6 - P7	0.981	0.922	0.059	6.0
P7 - P8	9.498	9.331	0.167	1.8
P8 - P9	9.169	9.009	0.160	1.7
P8 - P17	14.667	14.948	0.281	1.9
P10 - P11	1.214	1.217	0.002	0.2
P10 - P13	0.457	0.457	0.000	0.1
P11 - P12	0.456	0.460	0.004	0.9
P12 - P13	1.215	1.183	0.032	2.6
P21 - P22	3.023	3.040	0.017	0.6
P21 - P25	0.288	0.302	0.014	4.7
P22 - P23	1.535	1.573	0.038	2.5
P23 - P24	3.015	3.053	0.038	1.3
P24 - P28	0.284	0.272	0.012	4.3

	Theodolite	Phoenix	dDist (m)	Error (%)
P25 - P26	2.757	2.763	0.006	0.2
P25 - P28	0.984	1.007	0.023	2.4
P26 - P27	0.835	1.011	0.176	21.0
P27 - P28	2.768	2.784	0.016	0.6
P29 - P30	2.008	1.981	0.027	1.4
P29 - P32	1.048	1.118	0.070	6.7
P30 - P31	1.067	1.077	0.010	0.9
P30 - P33	0.344	0.342	0.002	0.6
P31 - P32	2.020	2.001	0.019	0.9
P31 - P34	0.349	0.366	0.017	4.9
P33 - P34	1.395	1.411	0.016	1.2
P35 - P36	2.013	2.011	0.002	0.1
P35 - P38	1.083	1.126	0.043	4.0
P36 - P37	1.070	1.107	0.036	3.4
P36 - P39	0.350	0.329	0.022	6.2
P37 - P38	2.010	2.010	0.000	0.0
P37 - P40	0.349	0.370	0.020	5.8
P39 - P40	1.420	1.436	0.016	1.1

Absolute	dX (m)	dY (m)	dZ (m)	dPos (m)	dDist (m)	Error (%)
Maximum	0.541	0.322	0.201	0.650	0.289	21.0
Mean	0.133	0.104	0.069	0.190	0.062	3.4
S.D. (pop.)	0.138	0.079	0.054	0.159	0.081	4.0

All Points

Absolute	dX (m)	dY (m)	dZ (m)	dPos (m)
Maximum	0.214	0.182	0.090	0.255
Mean	0.077	0.078	0.045	0.123
S.D. (pop.)	0.049	0.040	0.026	0.059

Removing first eight points from calculations

Table A1. This shows the calculated co-ordinates for the second survey detailed in Section 9.3 along with the absolute difference in position along each of the axes.

Table A2: Differences in distance between survey points

dDist	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P17	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40
P1	0.00	0.25	0.29	0.21	0.21	0.15	0.04	0.08	0.15	0.08	0.06	0.11	0.08	0.04	0.23	0.25	0.24	0.21	0.21	0.23	0.06	0.20	0.13	0.19	0.18	0.17	0.22	0.19	0.17	0.18	0.16	0.18	0.18	0.18
P2	0.25	0.00	0.29	0.23	0.27	0.26	0.25	0.37	0.37	0.33	0.33	0.36	0.32	0.11	0.44	0.50	0.47	0.40	0.42	0.47	0.28	0.41	0.33	0.41	0.39	0.35	0.45	0.40	0.35	0.38	0.35	0.31	0.38	0.36
P3	0.29	0.29	0.00	0.16	0.04	0.14	0.31	0.54	0.66	0.57	0.59	0.61	0.56	0.20	0.71	0.78	0.75	0.67	0.70	0.74	0.57	0.67	0.62	0.68	0.64	0.62	0.71	0.65	0.60	0.63	0.59	0.56	0.62	0.59
P4	0.21	0.23	0.16	0.00	0.05	0.04	0.12	0.38	0.50	0.41	0.43	0.45	0.40	0.09	0.56	0.62	0.59	0.52	0.54	0.58	0.41	0.52	0.47	0.52	0.49	0.47	0.55	0.50	0.46	0.48	0.44	0.42	0.47	0.45
P5	0.21	0.27	0.04	0.05	0.00	0.14	0.33	0.52	0.61	0.48	0.53	0.53	0.47	0.15	0.66	0.74	0.70	0.62	0.65	0.70	0.54	0.62	0.58	0.63	0.60	0.58	0.67	0.61	0.56	0.58	0.54	0.51	0.58	0.55
P6	0.15	0.26	0.14	0.04	0.14	0.00	0.06	0.38	0.49	0.36	0.40	0.41	0.36	0.06	0.53	0.60	0.57	0.49	0.52	0.57	0.41	0.50	0.47	0.51	0.48	0.46	0.54	0.49	0.44	0.46	0.42	0.40	0.46	0.43
P7	0.04	0.25	0.31	0.12	0.33	0.05	0.00	0.17	0.29	0.21	0.22	0.24	0.20	0.02	0.34	0.39	0.38	0.31	0.33	0.36	0.21	0.31	0.28	0.32	0.30	0.29	0.35	0.31	0.28	0.30	0.27	0.25	0.29	0.29
P8	0.08	0.37	0.54	0.38	0.52	0.38	0.17	0.00	0.16	0.08	0.11	0.09	0.09	0.28	0.15	0.17	0.13	0.12	0.15	0.15	0.09	0.12	0.16	0.11	0.08	0.11	0.11	0.07	0.10	0.08	0.03	0.05	0.06	0.04
P9	0.15	0.37	0.66	0.50	0.61	0.49	0.29	0.16	0.00	0.07	0.06	0.04	0.07	0.21	0.00	0.02	0.06	0.04	0.02	0.02	0.13	0.05	0.02	0.02	0.02	0.05	0.03	0.04	0.02	0.01	0.04	0.07	0.03	0.05
P10	0.08	0.33	0.57	0.41	0.48	0.36	0.21	0.08	0.07	0.00	0.00	0.03	0.00	0.12	0.13	0.17	0.14	0.11	0.12	0.14	0.06	0.10	0.07	0.11	0.09	0.07	0.13	0.10	0.08	0.10	0.07	0.05	0.09	0.08
P11	0.08	0.33	0.59	0.43	0.53	0.40	0.22	0.11	0.06	0.00	0.00	0.00	0.00	0.13	0.12	0.17	0.14	0.10	0.11	0.14	0.07	0.09	0.07	0.11	0.09	0.07	0.13	0.09	0.08	0.09	0.07	0.04	0.08	0.07
P12	0.11	0.36	0.61	0.45	0.53	0.41	0.24	0.09	0.04	0.03	0.00	0.00	0.03	0.14	0.10	0.14	0.11	0.08	0.09	0.11	0.11	0.07	0.04	0.08	0.06	0.04	0.10	0.06	0.06	0.07	0.05	0.02	0.06	0.05
P13	0.08	0.32	0.56	0.40	0.47	0.36	0.20	0.09	0.07	0.00	0.00	0.03	0.00	0.12	0.14	0.17	0.14	0.11	0.12	0.14	0.06	0.11	0.07	0.11	0.09	0.08	0.13	0.09	0.08	0.10	0.08	0.05	0.09	0.08
P17	0.04	0.11	0.20	0.09	0.15	0.06	0.02	0.28	0.21	0.12	0.13	0.14	0.12	0.00	0.20	0.17	0.13	0.19	0.19	0.14	0.03	0.18	0.21	0.20	0.19	0.16	0.18	0.17	0.18	0.20	0.17	0.15	0.18	0.16
P21	0.23	0.44	0.71	0.56	0.66	0.53	0.34	0.15	0.00	0.13	0.12	0.10	0.14	0.20	0.00	0.02	0.06	0.02	0.01	0.01	0.11	0.04	0.03	0.01	0.01	0.04	0.02	0.03	0.01	0.00	0.03	0.05	0.02	0.04
P22	0.25	0.50	0.78	0.62	0.74	0.60	0.39	0.17	0.02	0.17	0.17	0.14	0.17	0.17	0.02	0.00	0.04	0.02	0.02	0.03	0.20	0.03	0.07	0.03	0.02	0.00	0.02	0.00	0.02	0.04	0.00	0.02	0.02	0.00
P23	0.24	0.47	0.75	0.59	0.70	0.57	0.38	0.13	0.06	0.14	0.14	0.11	0.14	0.13	0.06	0.04	0.00	0.04	0.05	0.01	0.17	0.04	0.05	0.06	0.06	0.02	0.06	0.04	0.05	0.08	0.04	0.01	0.05	0.04
P24	0.21	0.40	0.67	0.52	0.62	0.49	0.31	0.12	0.04	0.11	0.10	0.08	0.11	0.19	0.02	0.02	0.04	0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.01	0.01	0.03	0.01	0.03
P25	0.21	0.42	0.70	0.54	0.65	0.52	0.33	0.15	0.02	0.12	0.11	0.09	0.12	0.19	0.01	0.02	0.05	0.01	0.00	0.01	0.09	0.02	0.04	0.00	0.00	0.02	0.01	0.02	0.00	0.01	0.01	0.03	0.01	0.03
P26	0.23	0.47	0.74	0.58	0.70	0.57	0.36	0.15	0.02	0.14	0.14	0.11	0.14	0.14	0.01	0.03	0.01	0.00	0.01	0.00	0.18	0.01	0.10	0.06	0.05	0.04	0.04	0.03	0.06	0.07	0.04	0.01	0.05	0.03
P27	0.06	0.28	0.57	0.41	0.54	0.41	0.21	0.09	0.13	0.06	0.07	0.11	0.06	0.03	0.11	0.20	0.17	0.01	0.09	0.18	0.00	0.02	0.18	0.25	0.24	0.18	0.24	0.22	0.22	0.26	0.23	0.18	0.24	0.22
P28	0.20	0.41	0.67	0.52	0.62	0.50	0.31	0.12	0.05	0.10	0.09	0.07	0.11	0.18	0.04	0.03	0.04	0.01	0.02	0.01	0.02	0.00	0.04	0.00	0.01	0.00	0.02	0.01	0.02	0.02	0.00	0.02	0.00	0.02
P29	0.13	0.33	0.62	0.47	0.56	0.47	0.28	0.16	0.02	0.07	0.07	0.04	0.07	0.21	0.03	0.07	0.05	0.02	0.04	0.10	0.18	0.04	0.00	0.03	0.00	0.07	0.03	0.01	0.05	0.01	0.04	0.09	0.02	0.06
P30	0.19	0.41	0.68	0.52	0.63	0.51	0.32	0.11	0.02	0.11	0.11	0.08	0.11	0.20	0.01	0.03	0.06	0.01	0.00	0.06	0.25	0.00	0.03	0.00	0.01	0.00	0.00	0.03	0.00	0.01	0.02	0.04	0.01	0.03
P31	0.18	0.39	0.64	0.49	0.60	0.48	0.30	0.08	0.02	0.09	0.09	0.06	0.09	0.19	0.01	0.02	0.05	0.00	0.00	0.05	0.24	0.01	0.00	0.01	0.00	0.01	0.00	0.02	0.02	0.02	0.01	0.02	0.00	0.03
P32	0.17	0.35	0.62	0.47	0.58	0.46	0.29	0.11	0.05	0.07	0.07	0.04	0.08	0.16	0.04	0.00	0.02	0.02	0.02	0.04	0.18	0.00	0.07	0.00	0.02	0.00	0.00	0.01	0.02	0.02	0.01	0.02	0.02	0.01
P33	0.22	0.45	0.71	0.55	0.67	0.54	0.35	0.11	0.03	0.13	0.13	0.10	0.13	0.18	0.02	0.02	0.06	0.02	0.01	0.04	0.24	0.02	0.03	0.00	0.00	0.00	0.00	0.02	0.01	0.03	0.01	0.03	0.00	0.02
P34	0.19	0.40	0.65	0.50	0.61	0.49	0.31	0.07	0.04	0.10	0.09	0.06	0.09	0.17	0.03	0.00	0.04	0.02	0.02	0.03	0.22	0.01	0.01	0.03	0.02	0.01	0.02	0.00	0.02	0.04	0.00	0.01	0.02	0.00
P35	0.17	0.35	0.60	0.46	0.56	0.44	0.26	0.10	0.02	0.08	0.06	0.06	0.06	0.18	0.01	0.02	0.05	0.01	0.00	0.06	0.22	0.02	0.05	0.00	0.02	0.02	0.01	0.02	0.00	0.00	0.00	0.04	0.01	0.02
P36	0.18	0.38	0.63	0.48	0.58	0.46	0.30	0.08	0.01	0.10	0.09	0.07	0.10	0.20	0.00	0.04	0.08	0.01	0.01	0.07	0.26	0.02	0.01	0.01	0.02	0.02	0.03	0.04	0.00	0.00	0.04	0.04	0.02	0.04
P37	0.18	0.35	0.59	0.44	0.54	0.42	0.27	0.03	0.04	0.07	0.07	0.05	0.08	0.17	0.03	0.00	0.04	0.01	0.02	0.04	0.23	0.00	0.04	0.02	0.01	0.01	0.01	0.01	0.00	0.04	0.00	0.00	0.01	0.02
P38	0.16	0.31	0.56	0.42	0.51	0.40	0.25	0.05	0.07	0.05	0.04	0.02	0.05	0.15	0.05	0.02	0.01	0.03	0.04	0.01	0.18	0.02	0.09	0.04	0.02	0.02	0.03	0.02	0.04	0.04	0.00	0.00	0.01	0.02
P39	0.18	0.38	0.62	0.47	0.58	0.46	0.29	0.06	0.03	0.09	0.06	0.06	0.09	0.18	0.02	0.02	0.05	0.01	0.01	0.05	0.24	0.00	0.02	0.01	0.00	0.02	0.00	0.02	0.01	0.02	0.01	0.01	0.00	0.02
P40	0.18	0.36	0.59	0.45	0.55	0.43	0.29	0.04	0.05	0.08	0.07	0.05	0.08	0.16	0.04	0.00	0.04	0.03	0.03	0.03	0.22	0.02	0.06	0.03	0.02	0.01	0.02	0.00	0.02	0.04	0.02	0.02	0.02	0.00

Table A2. This shows the three-dimensional difference in distance between points surveyed during the second test detailed in Section 9.3. For example, between points P1 and P2 the distance measured using the Phoenix approach is 0.25m different to that measured by theodolite. The distances in red are those above the average difference of 0.190m.

Table A3: Percentage differences in distance between survey points

%	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P17	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40
P1	-	3.56	3.34	2.09	1.89	1.19	0.38	0.42	0.81	0.64	0.57	0.80	0.63	0.17	1.18	1.34	1.26	1.07	1.11	1.18	0.30	1.06	0.65	0.99	0.94	0.85	1.14	0.99	0.86	0.93	0.89	0.79	0.94	0.92
P2	3.56	-	5.34	4.10	3.53	3.22	3.49	2.70	3.00	5.39	4.46	4.83	5.24	0.53	3.54	4.13	3.85	3.23	3.42	3.86	2.35	3.25	2.68	3.33	3.09	2.77	3.59	3.14	2.62	2.89	2.60	2.29	2.92	2.62
P3	3.34	5.34	-	10.81	1.67	4.07	10.54	4.38	4.52	7.02	6.51	6.50	6.64	1.04	4.89	5.91	5.61	4.57	4.80	5.59	4.31	4.60	4.32	5.01	4.68	4.26	5.30	4.77	4.03	4.43	4.02	3.64	4.47	4.09
P4	2.09	4.10	10.81	-	2.51	1.68	7.21	3.53	3.75	5.76	5.44	5.42	5.36	0.48	4.14	5.22	4.91	3.82	4.04	4.87	3.46	3.84	3.59	4.27	3.93	3.52	4.58	4.02	3.31	3.70	3.29	2.91	3.72	3.36
P5	1.89	3.53	1.67	2.51	-	11.01	20.73	4.81	4.15	5.50	5.52	5.40	5.16	0.85	4.45	5.71	5.41	4.14	4.37	5.36	4.16	4.17	4.04	4.78	4.44	3.93	5.09	4.55	3.70	4.20	3.77	3.31	4.24	3.67
P6	1.19	3.22	4.07	1.68	11.01	-	6.02	3.98	3.53	4.45	4.61	4.49	4.15	0.36	3.84	5.10	4.79	3.53	3.76	4.72	3.46	3.55	3.45	4.15	3.80	3.33	4.46	3.91	3.13	3.59	3.16	2.74	3.61	3.26
P7	0.38	3.49	10.54	7.21	20.73	6.02	-	1.76	2.19	2.81	2.82	2.85	2.60	0.10	2.55	3.48	3.28	2.31	2.47	3.14	1.80	2.32	2.19	2.73	2.50	2.18	3.02	2.60	2.09	2.42	2.11	1.79	2.40	2.22
P8	0.42	2.70	4.38	3.53	4.81	3.98	1.76	-	1.74	0.96	1.32	1.05	0.98	1.92	1.67	2.69	1.98	1.27	1.63	2.28	1.36	1.32	1.86	1.59	1.07	1.18	1.68	0.95	1.00	1.01	0.38	0.45	0.72	0.41
P9	0.81	3.00	4.52	3.75	4.15	3.53	2.19	1.74	-	1.01	1.14	0.69	1.08	1.23	0.75	0.50	1.77	2.19	5.02	0.57	4.32	3.44	0.56	0.42	0.47	1.26	0.67	0.84	0.48	0.21	0.60	1.06	0.48	0.77
P10	0.64	5.39	7.02	5.76	5.50	4.45	2.81	0.96	1.01	-	0.20	2.32	0.11	0.69	1.98	2.85	2.31	1.56	1.80	2.33	1.05	1.54	0.96	1.63	1.32	1.01	2.00	1.32	1.03	1.27	0.88	0.57	1.17	0.88
P11	0.57	4.46	6.51	5.44	5.52	4.61	2.82	1.32	1.14	0.20	-	0.95	0.30	0.78	2.24	3.61	2.73	1.67	2.03	2.93	1.55	1.66	1.19	1.92	1.45	1.05	2.34	1.40	1.05	1.37	0.84	0.48	1.22	0.83
P12	0.80	4.83	6.50	5.42	5.40	4.49	2.85	1.05	0.69	2.32	0.95	-	2.62	0.83	1.90	2.83	2.06	1.35	1.67	2.22	2.20	1.32	0.75	1.39	1.04	0.70	1.77	0.99	0.79	1.04	0.59	0.27	0.89	0.57
P13	0.63	5.24	6.64	5.36	5.16	4.15	2.60	0.98	1.08	0.11	0.30	2.62	-	0.69	2.11	2.80	2.27	1.67	1.92	2.32	1.05	1.65	0.98	1.63	1.32	1.04	1.98	1.31	1.05	1.27	0.88	0.59	1.17	0.86
P17	0.17	0.53	1.04	0.48	0.85	0.36	0.10	1.92	1.23	0.69	0.78	0.83	0.69	-	1.18	1.07	0.90	1.18	1.13	0.93	0.21	1.10	1.46	1.45	1.51	1.16	1.35	1.37	1.38	1.72	1.61	1.25	1.60	1.51
P21	1.18	3.54	4.89	4.14	4.45	3.84	2.55	1.67	0.75	1.98	2.24	1.90	2.11	1.18	-	0.57	1.66	1.60	4.73	0.48	3.76	2.92	1.11	0.24	0.28	0.96	0.54	0.68	0.22	0.03	0.44	0.87	0.31	0.63
P22	1.34	4.13	5.91	5.22	5.71	5.10	3.48	2.69	0.50	2.85	3.61	2.83	2.80	1.07	0.57	-	2.47	0.64	0.50	7.70	17.80	0.81	1.99	1.09	0.49	0.08	0.68	0.01	0.44	0.85	0.07	0.33	0.34	0.01
P23	1.26	3.85	5.61	4.91	5.41	4.79	3.28	1.98	1.77	2.31	2.73	2.06	2.27	0.90	1.66	2.47	-	1.27	1.43	0.82	30.67	1.30	2.19	6.04	2.61	0.76	6.64	1.71	1.20	2.38	0.95	0.17	1.74	0.82
P24	1.07	3.23	4.57	3.82	4.14	3.53	2.31	1.27	2.19	1.56	1.67	1.35	1.67	1.18	1.60	0.64	1.27	-	0.87	0.01	0.20	4.31	1.90	0.41	0.14	0.83	0.70	0.63	0.28	0.15	0.26	0.66	0.16	0.54
P26	1.11	3.42	4.80	4.04	4.37	3.76	2.47	1.63	5.02	1.80	2.03	1.67	1.92	1.13	4.73	0.50	1.43	0.87	-	0.24	3.01	2.37	1.70	0.02	0.04	0.65	0.35	0.48	0.06	0.18	0.26	0.67	0.11	0.47
P26	1.18	3.86	5.59	4.87	5.36	4.72	3.14	2.28	0.57	2.33	2.93	2.22	2.32	0.93	0.48	7.70	0.82	0.01	0.24	-	21.01	0.22	3.21	2.61	1.49	0.90	2.12	0.85	1.13	1.57	0.83	0.20	1.05	0.50
P27	0.30	2.35	4.31	3.46	4.16	3.46	1.80	1.36	4.32	1.05	1.55	2.20	1.05	0.21	3.76	17.80	30.67	0.20	3.01	21.01	-	0.58	7.29	17.54	9.62	5.48	17.73	8.21	5.15	7.04	4.71	3.53	6.64	4.41
P28	1.06	3.25	4.60	3.84	4.17	3.55	2.32	1.32	3.44	1.54	1.66	1.32	1.65	1.10	2.92	0.81	1.30	4.31	2.37	0.22	0.58	-	2.88	0.11	0.14	0.20	0.47	0.37	0.61	0.41	0.03	0.36	0.09	0.31
P29	0.65	2.68	4.32	3.59	4.04	3.45	2.19	1.86	0.56	0.96	1.19	0.75	0.98	1.46	1.11	0.99	2.19	1.90	1.70	3.21	7.29	2.88	-	1.36	0.16	6.70	1.15	0.52	2.08	0.39	1.14	2.71	0.54	1.37
P30	0.99	3.33	5.01	4.27	4.78	4.15	2.73	1.59	0.42	1.63	1.92	1.39	1.63	1.45	0.24	1.09	6.04	0.41	0.02	2.61	17.54	0.11	1.36	-	0.90	0.05	0.59	2.24	0.15	0.55	0.70	1.03	0.56	0.85
P31	0.94	3.09	4.68	3.93	4.44	3.80	2.50	1.07	0.47	1.32	1.45	1.04	1.32	1.51	0.28	0.49	2.61	0.14	0.04	1.49	9.62	0.14	0.16	0.90	-	0.95	0.01	4.93	0.73	1.81	0.61	0.80	0.34	0.88
P32	0.85	2.77	4.26	3.52	3.93	3.33	2.18	1.18	1.26	1.01	1.05	0.70	1.04	1.16	0.96	0.08	0.76	0.83	0.65	0.90	5.48	0.20	6.70	0.05	0.95	-	0.01	0.43	1.99	1.03	0.40	0.84	0.81	0.16
P33	1.14	3.59	5.30	4.58	5.09	4.46	3.02	1.68	0.67	2.00	2.34	1.77	1.98	1.35	0.54	0.68	6.64	0.70	0.35	2.12	17.73	0.47	1.15	0.59	0.01	0.01	-	1.17	0.30	1.02	0.34	0.76	0.01	0.45
P34	0.99	3.14	4.77	4.02	4.55	3.91	2.60	0.95	0.84	1.32	1.40	0.99	1.31	1.37	0.68	0.01	1.71	0.63	0.48	0.85	8.21	0.37	0.52	2.24	4.93	0.43	1.17	-	0.66	3.75	0.20	0.49	1.82	0.00
P35	0.86	2.62	4.03	3.31	3.70	3.13	2.09	1.00	0.48	1.03	1.05	0.79	1.05	1.38	0.22	0.44	1.20	0.28	0.06	1.13	5.15	0.61	2.08	0.15	0.73	1.99	0.30	0.66	-	0.08	0.01	4.00	0.59	0.78
P36	0.93	2.89	4.43	3.70	4.20	3.59	2.42	1.01	0.21	1.27	1.37	1.04	1.27	1.72	0.03	0.85	2.38	0.15	0.18	1.57	7.04	0.41	0.39	0.55	1.81	1.03	1.02	3.75	0.08	-	3.39	1.61	6.19	3.32
P37	0.89	2.60	4.02	3.29	3.77	3.16	2.11	0.38	0.60	0.88	0.84	0.59	0.88	1.61	0.44	0.07	0.95	0.26	0.26	0.63	4.71	0.03	1.14	0.70	0.61	0.40	0.34	0.20	0.01	3.39	-	0.02	0.80	5.84
P38	0.79	2.29	3.64	2.91	3.31	2.74	1.79	0.45	1.06	0.57	0.48	0.27	0.59	1.25	0.87	0.33	0.17	0.66	0.67	0.20	3.53	0.36	2.71	1.03	0.80	0.84	0.76	0.49	4.00	1.61	0.02	-	0.56	0.85
P39	0.94	2.92	4.47	3.72	4.24	3.61	2.40	0.72	0.48	1.17	1.22	0.89	1.17	1.60	0.31	0.34	1.74	0.16	0.11	1.05	6.64	0.09	0.54	0.56	0.34	0.81	0.01	1.82	0.59	6.19	0.80	0.56	-	1.12
P40	0.92	2.62	4.09	3.36	3.67	3.26	2.22	0.41	0.77	0.88	0.83	0.57	0.86	1.51	0.63	0.01	0.82	0.54	0.47	0.50	4.41	0.31	1.37	0.85	0.88	0.16	0.45	0.00	0.78	3.32	5.84	0.85	1.12	-

Table A3. This shows the three-dimensional difference as a percentage in distance between points surveyed during the second test detailed in Section 9.3.